

# Werkcollege 3: Cookies en sessions

## Oefening 1: Bijlagen uploaden

Maak een eenvoudige webapplicatie aan met twee schermen:

- Het eerste scherm toont een formulier waarmee de gebruiker een nieuw image bestand (enkel jpeg of png) kan uploaden naar de server. Denk eraan dat de map op je webserver waar je de image file in zal opslaan beschrijfbaar moet zijn. De gebruiker krijgt bij het succesvol uploaden van de foto het tweede scherm te zien.
- Het tweede scherm toont de afbeelding die de gebruiker heeft geupload door middel van het formulier van het eerste scherm. Het biedt ook de mogelijkheid om terug te gaan naar het eerste scherm om een nieuwe foto te uploaden.

## Oefening 2: Eenvoudige Cookies:

Maak een eenvoudige webapplicatie met 2 schermen:

- Homescherm:
  - Toont een welkomsttekst (verzin er zelf één) in één van de volgende 3 talen:
    - Nederlands
    - Frans
    - Engels
  - De welkomsttekst wordt getoond in de taal die werd ingesteld door de gebruiker. De instellingen worden opgeslagen in een cookie dat via het instellingen scherm kan worden aangemaakt. Indien er geen cookie werd aangemaakt, is de default taal Nederlands.
  - Bevat een link naar het “instellingen”scherm
- Instellingen scherm:
  - Bevat een formulier waarmee de gebruiker een voorkeurstaal kan instellen (Nederlands, Frans of Engels). De keuze wordt gemaakt door middel van radiobuttons.
  - Wanneer een voorkeurstaal wordt aangeduid, wordt deze preferentie opgeslagen in een tijdelijke cookie.

Test de webapplicatie uit door een voorkeurstaal te kiezen. Sluit daarna je browser af. Werd de voorkeurstaal onthouden? Waarom (niet)?

### Oefening 3: Uitgebreide cookies

Kopieer de oplossing van oefening 2 naar een nieuw project en breid ze uit met de volgende functionaliteiten:

- Homescherm:
  - De welkomsttekst wordt nog steeds weergegeven in de ingestelde taal, maar bevat nu echte naam van de bezoeker indien die gekend is (vb: “Welkom Maarten Heylen”). Indien de naam van de bezoeker niet gekend is, wordt de bezoeker aangesproken als “bezoeker” (vb: “Welkom bezoeker”). De **naam van de bezoeker** wordt bijgehouden in een cookie die wordt ingesteld via het instellingenschermb.
  - De gebruiker heeft de mogelijkheid om het homescherm te customizen qua uitzicht. Via het instellingen scherm kan de gebruiker een kleur kiezen. Deze kleur zal gebruikt worden om de achtergrond van het homescherm mee te vullen. Indien geen kleur werd ingesteld door de gebruiker, is de **achtergrondkleur** wit. De ingestelde kleur wordt bijgehouden in een cookie die wordt ingesteld via het instellingenschermb.
  - Het huidige tijdstip wordt weergegeven op het scherm (moet enkel refreshen bij het herladen van de pagina => server side code, niet client side). Aangezien de gebruikers van onze website verspreid zitten over heel de wereld, willen we de voor de gebruiker relevante lokale tijd laten zien. Het weergegeven tijdstip houdt rekening met de **tijdszone** die de gebruiker heeft ingesteld.
- Instellingenschermb
  - Er wordt een tekstveld toegevoegd waarin de gebruiker zijn/haar naam kan invullen. Deze informatie wordt opgeslagen in een tijdelijke cookie.
  - Er wordt een keuzemenu (listbox) aangeboden waaruit de gebruiker een achtergrondkleur voor de website kan kiezen. Deze informatie wordt opgeslagen in een semi-permanente cookie die voor een week wordt bijgehouden.
  - Er wordt een keuzemenu (dropdown) aangeboden waaruit de gebruiker een tijdszone kan kiezen voor de website. Deze informatie wordt opgeslagen in een semi-permanente cookie die tot 31 december 2014, 23u59 wordt bijgehouden.

Test de webapplicatie uit door enkele instellingen te maken. Sluit daarna je browser af. Welke gegevens werden onthouden? Welke niet? Waarom (niet)?

Wis de cookies voor de webapplicatie uit je browser en observeer hoe de webapplicatie reageert.

### Oefening 4: Sessions

Kopieer de oplossing van oefening 3 naar een nieuw project en pas ze aan.

Cookie informatie wordt bij elke request van de client naar de server gestuurd. Aangezien

we in oefening 3 al redelijk wat informatie over de gebruiker bijhouden, zorgt dit voor extra netwerkverkeer. Aangezien onze webapplicatie ook veel door mobiele gebruikers wordt gebruikt – en hun bandbreedte beperkt is – heeft onze opdrachtgever bepaald dat we een oplossing moeten zoeken waarmee er minder informatie heen en weer wordt gestuurd bij elke request. Het is om deze reden dat we oefening 3 zullen hermaken zodat deze met sessions werkt in plaats van met cookies.

Maak de nodige aanpassingen en test de webapplicatie uit. Zorg ervoor dat alle functionaliteiten nog werken.

## Oefening 5: Sessions en objecten

Kopieer de oplossing van oefening 4 naar een nieuw project en pas ze aan.

We hebben een heel aantal instellingen die we willen bijhouden. Om hier wat meer structuur in te brengen zullen we een klasse schrijven met de volgende attributen:

- Taalkeuze
- Bezoekersnaam
- Achtergrondkleur
- Tijdszone

We zullen de instellingen voor een gebruiker opslaan in een instantie van deze klasse. Dit geeft ons naar de toekomst meer mogelijkheden (methoden toevoegen, ...). Het is de instantie van deze klasse die we nu zullen opslaan in het session object in plaats van de losse instellingen. Merk op dat je objecten moet serialiseren vooraleer je ze kan bewaren.

Maak de nodige aanpassingen en test de webapplicatie uit. Zorg ervoor dat alle functionaliteiten nog werken.

## Oefening 6: Een eenvoudig login systeem met sessies

Zorg voor enkele 'beveiligde' pagina's met behulp van sessies. Maak enkele verschillende pagina's aan:

- login.php: Deze pagina wordt gebruikt om te kunnen inloggen. Ze toont een formulier waar de gebruiker een gebruikersnaam en wachtwoord kan ingeven. Wanneer dit gebeurd is, moet je ervoor zorgen dat de gebruiker gedurende de sessie steeds op de geheime pagina's kan. De gebruikersnaam en wachtwoord worden hardcoded. Wanneer de loginpagina wordt opgevraagd wanneer de gebruiker reeds is ingelogd, wordt de pagina geheim1.php getoond.
- geheim1.php: Een eerste pagina dat niet toegankelijk mag zijn als de gebruiker zich niet eerst ingelogd heeft. Indien niet ingelogd, toon je de pagina login.php aan de gebruiker.
- geheim2.php: Een tweede pagina dat niet toegankelijk mag zijn als de gebruiker zich niet heeft ingelogd. Indien niet ingelogd, toon je de pagina login.php aan de

gebruiker.

- `logout.php`: Deze pagina gaat de sessie verwijderen zodat de gebruiker weer eerst moet inloggen om aan de twee beveiligde pagina's te kunnen. Enkele dingen die je kan uitproberen wanneer de oefening werkt:
  - Surf naar een andere pagina, nadat je bent ingelogd en probeer dan nadien direct de beveiligde pagina's te lezen.
  - Sluit de browser nadat je bent ingelogd en probeer opnieuw de beveiligde pagina's te bekijken.
  - Log uit en probeer met de backtoets van de browser terug naar de beveiligde pagina te surfen.

Zorg voor de nodige links tussen de verschillende pagina's. Probeer zoveel mogelijk van de functionaliteit te centraliseren zodat de code herbruikt kan worden door middel van includes.