Hanzehogeschool Groningen
University of Applied Sciences

Instituut voor Communicatie, Media & IT

Thema 4.1 I:
Web Application Development

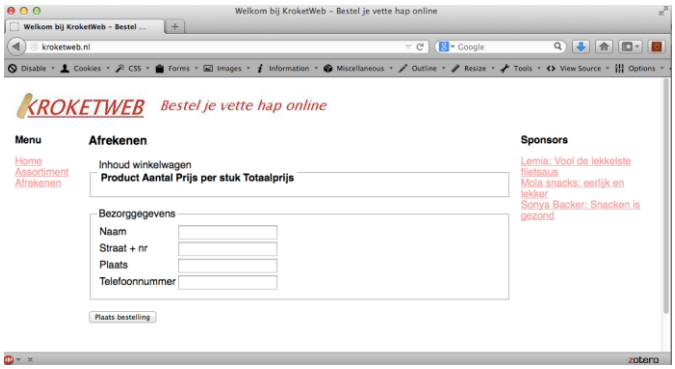
Opgaven week 2

1. Inleiding

In deze week wordt er ingegaan op de nieuwste specificaties van html en css, waarvoor in de theorieles een uitgebreide basis gelegd is. Verder staan javascript en javascript-frameworks centraal (zowel client- als server-side).

Opgave 1: html5 en css3

Voor deze opgave moet gebruik gemaakt worden van een moderne browser; zie http://caniuse.com/ voor een overzicht van welke tags door welke browsers ondersteund worden. Op het moment is Google Chrome de browser die het meeste ondersteuning biedt.



1. Download kroketweb.zip van Blackboard. Deze zip bevat een skelet van een site waarmee het mogelijk moet zijn snacks online te bestellen. Pak de zip uit en zorg ervoor dat deze via de browser te bereiken is.

2. Maak gebruik van de mogelijkheden van html5 en css3 om de volgende eigenschappen te implementeren:

- ✓ De tekst op de home-pagina moet over twee kolommen verdeeld worden, met veertig pixels ruimte en een dunne lijn tussen de beide kolommen;
- ✓ Gebruik aside en nav om de sponsor- en navigatiehyperlinks van elkaar te onderscheiden; verander ook de bijhorende css;
- ✓ De lijn onder de hyperlinks naar de sponsors aan de rechterkant moet verdwijnen en als een bezoeker over één van deze hyperlinks gaat, moet er een gekleurde achtergrond achter de letters komen;
- ✓ Het menu aan de linkerkant moet duidelijk aangeven welke pagina momenteel getoond wordt;
- ✓ Geef de rijen van de tabel op de assortiment-pagina alternerende kleuren, zonder de class van die rij te veranderen;

Maak gebruik van standaard javascript (dus zonder gebruik te maken van een framework) om de volgende functionaliteit te implementeren:

- ✓ Op de Assortiment-pagina kan de bezoeker een geheel getal in het veld achter de snack invoeren;
- ✓ Wanneer de bezoeker dit veld verlaat moet de snack, de prijs en het bestelde aantal links in het winkelmandje-overzicht terecht komen;

- ✓ Onderaan de tabel met het winkelwagentje moet de total prijs komen te staan;
- ✓ Dit winkelwagentje moet ook terugkomen op de Afrekenen-pagina;
- ✓ Zorg ervoor dat de relevante controles in de verschillende velden van de bezorggegevens op de Afrekenen-pagina worden uitgevoerd;

- ✓ Maak gebruik van sessionStorage om ervoor te zorgen dat een bezoeker bij een volgend bezoek haar bestelling nog kan zien.

3. Download kroketweb.sql van Blackboard, laad dit in een database en zorg ervoor dat de gegevens op de Assortiment-pagina uit deze database gehaald worden. Zorg er ook voor dat het assortiment over meerdere pagina's verdeeld weergegeven wordt, met de pagina-links eronder.

4. Handel de bestelling server-side af:

- ✓ de gegevens van de klant en de bestelling komen in de database;
- ✓ het mandje van de bezoeker wordt leeggemaakt;
- ✓ de bezoeker krijg een melding te zien dat haar bestelling geplaatst is

5. Demonstreer de werking en de programmacode aan de practicumdocent.

Opgave 2: memory

Deze opgave gaat over het bekende spel Memory. Bij dit spel wordt een even aantal kaarten omgekeerd op tafel legt, waarbij elke kaart twee keer voorkomt.Elke beurt draait de speler twee kaarten om. Als op deze kaarten dezelfde afbeelding staat, mag de speler de kaarten houden en nog twee kaarten omdraaien. Als er twee verschillende afbeeldingen op de kaarten staan, gaat de beurt voorbij en mag de andere speler. Het spel eindigt wanneer er geen kaarten meer op tafel liggen; degene met de meeste kaarten heeft gewonnen.

Een variant van dit spel is een solitaire: hier gaat het er om zo snel mogelijk het speelveld leeg te spelen. Het is deze variant die in deze opgave gemaakt moet worden. Maak hierbij gebruik van de toegewezen JavaScript-wrapper (JQuery of MooTools).

1. Maak een pagina in html5 met een aardige css3 vormgeving, waarin de volgende functionele eisen worden geïmplementeerd:

- ✓ Er is een speelveld van zes keer zes cellen. In elke cel zit een letter verborgen;
- ✓ De speler ziet alleen asterisk (*) in elke cel staan;
- ✓ Via een *slider* kan de speler de toontijd van een beurt instellen
- ✓ Wanneer de speler op een cel klikt, wordt de letter die in deze cel staat getoond (als een speler op een al omgedraaide letter klikt gebeurt er niets);
- ✓ Bij het tonen van de eerste cel van een beurt gaat de toontijd lopen; de verlopen tijd wordt weergegeven;

- ✓ Als beide cellen verschillende letters bevatten, worden beide kaarten net zo lang weergegeven tot de toontijd voorbij is; hierna wordt de asterisk weer weergegeven;
- ✓ Als beide letters gelijk zijn, stopt het lopen van de toontijd en worden de letters in deze cellen permanent getoond;
- ✓ De allereerste keer dat de speler een cel klikt, wordt de starttijd van het spel opgeslagen.
- ✓ Als alle cellen zijn getoond, wordt de speler gefeliciteerd en de speeltijd weergegeven.

2. Demonstreer de werking en de programmacode aan de practicumdocent.

Opgave 3: server side javascript

In deze opgave moet het server-side deel van het memory-spel uit de vorige opgave gemaakt worden. Indien dit spel niet is afgekomen, volstaat het ook om van een simulatie van de noodzakelijke functionaliteit gebruik te maken.

1. Bestudeer het artikel *Javascript conquers the server*, waarvan de link op Blackboard te vinden is. In dit artikel worden een aantal server side javascript engines met elkaar vergeleken. Maak een overzicht van deze engines en beschrijf de overeenkomsten en verschillen hiertussen.

2. Maak gebruik van NodeJS om de volgende functionaliteit aan het memory-spel uit de vorige opgave toe te voegen:

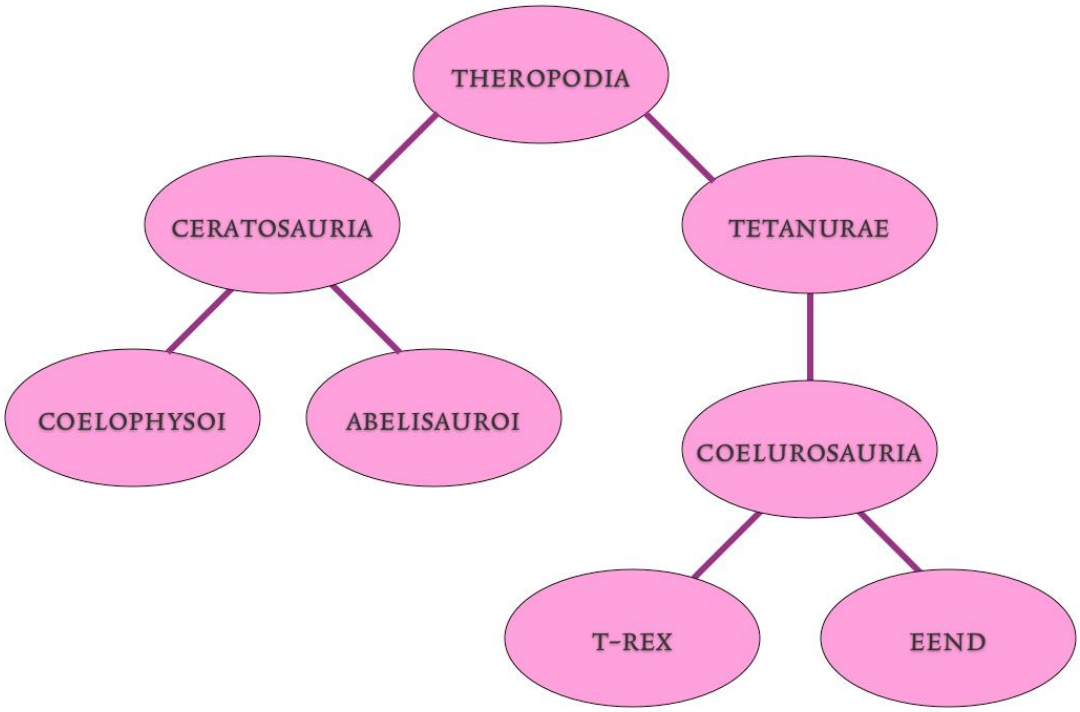
- ✓ Als het spel is afgelopen, wordt de speler de mogelijkheid geboden om haar naam in te voeren;
- ✓ Deze naam wordt, samen met de tijd die zij er over heeft gedaan, server side opgeslagen in een database;
- ✓ Op de pagina van het spel wordt ergens een top tien van behaalde scores weergegeven, waarbij de speler die het spel het snelst heeft opgelost bovenaan staat;
- ✓ Tijdens het speler wordt deze top tien continu bijgewerkt – dus als een andere speler in de top tien terecht is gekomen wordt dat direct in de top tien weergegeven (maak gebruik van de websocket.io module).

3. Demonstreer de werking en de code aan de practicumdocent.

Opgave 4: tree-traversal

1. Maak een classe Node in de toegewezen programmeertaal. Elke Node heeft een String naam en een methode performAction(). Deze methode geeft de naam van de Node weer en kijkt of deze Node siblings of childNodes heeft.

2. Maak een programma waarin deze classe Node wordt gebruikt om de volgende boomstructuur te implementeren:



3. Zorg ervoor dat van elke Node in deze boom de methode performAction() wordt uitgevoerd. Zorg er hierbij voor dat er vanuit dit programma maar één call naar één Node gedaan wordt en dat de verschillende Nodes de rest doen.
4. Demonstreer de werking aan de practicumdocent.
5. Geef aan of dit programma *breath-first* of *depth-first* en *top-down* of *bottom-up* is en beargumenteer deze keuze.
6. Beschrijf welke functies tree-traversal in programmeertalen in het algemeen heeft en waarom de ‘losse grammatica’ van html hierbij een probleem is.
