

Tillståndslös miljö och uppkopplingslös datakommunikation

I laboration 4 skall ni skapa lösningar för att rätt fil inkluderas när användaren trycker på någon av länkarna i vänstermenyn. Ni skall också skapa själva lösningarna (med gömda fält, modifierad url, kakor och sessioner) för att kunna skriva ut, slumpa och sortera 100 tal utan att "tappa bort" talen mellan körningarna (tillståndslös miljö). Ni skall därför genomföra punkterna nedan. Observera att uppgiften är sin tur uppdelad i flertalet delmoment och denna laboration är den största av de sju som skall mynna ut i en fungerande webbapplikation med testdokumentation.

A.

1. Skapa katalogen site som en underkatalog till incl.
2. Skapa katalogen src på samma nivå som övriga (incl, images osv.).
3. Skapa filen demo.php och spara den i incl/site
4. Ta bort hela tabellen från content.php. Spara en rad med tillhörande kolumner men kommentera bort dessa.
5. Inkludera demo.php under div:en med id=contentsection.
6. Flytta div:en med id=buttonlink (inklusive hela dess innehåll) till demo.php
7. Provkör
8. Validera
9. Skapa filen MyArray.php och placera den i src katalogen (dess innehåll kommer vi skapa på en av föreläsningarna alt. räknestuga 1).
10. Validera

B.

1. Skapa nu filerna default.php, hidden.php, modurl.php, cookie.php och session.php och spara dessa i incl/site
2. I respektive fil skall ni skriva ut filens namn. använd echo("filnamn.php");
3. Öppna nav.php och låt samtliga länkar peka på content.php (huvudfilen).
4. Provkör och låt muspekare "sväva" (hover) över länkarna till vänster. Titta längst ner till vänster (Firefox) där ni skall nu se var dessa pekar.
5. Gå tillbaka till nav.php och ändra länkarna enligt följande (modifierad url):
 - a. för länken med texten 'Gömda fält' skall ni ange content.php?p=h
 - b. för länken med texten 'Modifierad url' skall ni ange content.php?p=m
 - c. för länken med texten 'Kakor' skall ni ange content.php?p=c
 - d. för länken med texten 'Sessioner' skall ni ange content.php?p=s
6. Spara och provkör och "sväva" (hover) muspekaren över länkarna. Någon ändring?
7. Validera

C.

1. Öppna huvudfilen (content.php) och kommentera bort allt som finns i div:en med id=contentsection.
2. Ni skall nu ta reda på och inkludera rätt fil för den länk användaren har tryckt på.
3. Detta skall göras i div:en med id=contentsection.
4. Börja med att kontrollera om variabeln p har ett värde och om så är fallet kontrollera vilket värde den har Tips! isset() och \$_GET[]!
5. Är det h skall ni tilldela variabeln \$filetoinclude värdet hidden.php och variabeln \$method hidden och motsvarande för om variabeln har värden m, c eller s (modurl.php, cookie.php och session.php).
6. Har inte variabeln p något värde skall ni istället använda er av default.php och strängen "ingen metod vald ännu!"
7. Avslutningsvis skall ni inkludera MyArray.php och den filen vars namn finns i variabeln filetoinclude.
8. Testkör och validera.

D-G.

Skapa nu lösningar för problemet med tillståndslös miljö genom att skriva kod i hidden.php (input type="hidden", \$_POST), modifierad url (modurl.php?p=m&action=slumpa&strArray=..., \$_GET), kakor (setcookie(), \$_COOKIE) och session.php (session_start(), \$_SESSION).

H.

Avslutningsvis skall ni också testa era lösningar genom att först manuellt gå igenom och testa samtliga genererade sidor (prova klickordningen Sortera, Slumpa och Sortera) och se till att ni inte får några fel någonstans). Webapplikationen skall alltså fungera utan några problem.

Ni skall också validera er genererade kod (kör både slumpa och sortera för samtliga metoder, högerklicka och kopiera och klistra in i validatorn) och göra ACID tester för Firefox och Chrome.

Det sista ni skall göra i laboration 4 är att använda browsershots.org för att generera skrämdumpar för ett antal olika webbläsare på olika plattformar samt validator.w3.org/checklink för att kontrollera att länkar fungerar som de skall. Använd valfri sida när ni testar dessa två men tänk på att ett flertal servrar har stängt möjligheten för mjukvara att automatiskt komma åt dess innehåll. För browsershots.com väljer ni 5 webbläsare för varje plattform. När ni väl får följande sida som talar om att er förfrågan håller på att exekvera skall ni också passa på att läsa lite om tjänsten. Observera att detta kan ta en stund för att samtliga valda webbläsarna skall ges en bild. När ni erhållit bilder för samtliga webbläsare skall ni spara dessa eftersom ni behöver bilderna för er slutgiltiga dokumentation (er sista laboration). När ni också fått svaret från checklink är ni klara med laboration 4.

När ni är färdiga med uppgiften skall ni skriva några "kom ihåg" rader om vad ni har gjort och hur ni har gjort det och klistra in detta som svar på uppgiften. Kom också ihåg att ange vilka som varit med och genomfört uppgiften (välj namnen från listan när ni lämnar in uppgiften). Ni kommer att behöva dessa "kom ihåg" anteckningar i samband med sista labben (skriv hellre en längre och detaljerad

beskrivning än en kort och innehållsfattig) då ni skall lämna in en fungerande webbapplikation med tillhörande testdokumentation.

Bra länkar för uppgiften:

- Markup Validation Service <http://validator.w3.org/>
- CSS Validation Service <http://jigsaw.w3.org/css-validator/>
- Unicorn – W3C's Unified Validator <http://validator.w3.org/unicorn/>
- W3Schools <http://www.w3schools.com/>
- PHP: Hypertext Preprocessor <http://www.php.net/>
- ACID <http://acid3.acidtests.org/>
- Skapa "skärmdumpar" <http://browsershots.org/>
- Kontrollera länkar <http://validator.w3.org/checklink>

Bra litteratur för uppgiften:

- Gilmore, W.J. (2010) *Beginning PHP and MySQL From Novice to Professional*, Apress.
- Di Lucca, G.A. & Fasolino, A.R. (2006) "Testing web-based applications: The state of the art and future trends", *Information and Software Technology*, 48, 1172-1186.
- Föreläsningsbilder