## TNMK30 Lab 3: En mycket enkel blogg

#### Niklas Rönnberg, Stefan Gustavson, ITN-LiTH 2017

Många webtjänster är inte bara *interaktiva* i den meningen att de har menyer och andra funktioner som svarar på användarens aktivitet, utan även *dynamiska*, vilket innebär att innehållet kan ändras, ofta via ett gränssnitt som också är webbaserat. Det här är faktiskt en av de största fördelarna med webben som den ser ut i dag: innehållet kan både betraktas och redigeras med samma verktyg – en vanlig webbläsare.

Vi ska nu göra en dynamisk webbtjänst i form av en enkel "blogg", en sida där man kan publicera korta inlägg. Vi börjar med en enkel lösning som fungerar men inte är så smidig: att lägga inläggen i en textfil som uppdateras via PHP genom att man lägger till nya poster i slutet av filen. I nästa laboration kommer du att skapa en bättre och smidigare lösning som bygger på samma grund men använder en databas för det dynamiska innehållet.

### Inkludera innehåll från annan fil

För att inkludera en fil i en annan fil finns PHP-funktionen "include". Om inläggen för bloggen ligger i filen "blog.txt" så kan de visas som en del av ett HTML-dokument med följande PHP-kod:

#### <?php include("blog.txt"); ?>

Texten från den inkluderade filen läggs in på det ställe i dokumentet där funktionsanropet görs. För att slutresultatet ska bli välstrukturerad HTML-kod så krävs att den inkluderade filen har korrekta och balanserade HTML-taggar, men den ska inte vara ett fullständigt HTML-dokument, utan bara innehålla vissa delar av koden för den totala sidan.

**Gör en enkel bloggsida som inkluderar en annan fil enligt ovan!** Få in bloggsidan i den layout och struktur som du redan har i den webbsida du skapat under de tidigare laborationerna i kursen. Filen "blog.txt" kan du skapa själv så länge, för att testa att själva visningen fungerar.

### Posta från formulär

Nu ska vi göra ett gränssnitt som gör det möjligt att posta nya inlägg i bloggen wia webben. För att skriva texten behövs ett formulär. **Skapa en HTML-sida med ett formulär** med tre fält: ett för namnet på den som skrivit inlägget, ett för rubriken på inlägget och ett för själva texten för inlägget. Ge fälten vettiga namn och lämpliga storlekar, och sätt rubriker intill dem på sidan så att användaren förstår vad de är till för. Namn och rubrik kan vara <input type="text">, medan inlägget bör vara av typen <textarea> för att kunna hantera längre texter. Gör också en "Submit"-knapp (<input type="submit">) så att man kan posta formuläret.

**Gör en testsida i PHP för att ta emot formulärdata** så att du ser att formuläret fungerar. Nedan ser du hur man i PHP tar emot data från ett formulär. Variabeln **\$\_POST** är en array som innehåller alla data från formuläret. Den här koden ska alltså ligga i den fil som pekas ut av attributet action i HTML-formuläret.

```
<?php
$author = $_POST['author']; // Namnet på formulärfälten bestäms
$heading = $_POST['heading']; // med attribut i HTML-koden för formuläret
$entry = $_POST['entry'];
print("<h1>$heading</h1>\n");
print("<em>$author</em>\n");
print("$entry\n");
?>
```

### Spara data

Data från formuläret behöver förstås sparas mer permanent. Som de flesta programmeringsspråk så kan PHP läsa och skriva textfiler. Det är litet pyssligt att ge webbservern rättigheter att ändra i filer på din

webbkatalog, och hur man gör beror helt på vilket filsystem och vilket operativsystem man kör, men för studentkatalogerna på ITN monterade i Windows-miljö gör du så här:

**Skapa filen som PHP ska kunna ändra**, till exempel genom att högerklicka på en tom yta i mappen där filen ska ligga och välja *New→Text Document*. Byt namn på filen till "blog.txt".

**Gör filen skrivbar för webservern.** Högerklicka på filen, välj *Properties*... och ta fram fliken *Sexurity*. Klicka på *Edit*..., välj *Everyone* och kryssa i *Write*. Det ska se ut som på bilden här bredvid. Klicka *OK*.

Nu har du gett tillåtelse för bland annat webservern att inte bara öppna och läsa filen "blog.txt", utan även ändra i den, till exempel via PHP. För att lägga till text i slutet av filen kan du använda följande kod:

// Append to file (a). Use "binary mode"
// (b) even for text files, as recommended
// by the PHP documentation.
\$blogfile = fopen("blog.txt", "ab");
fwrite(\$blogfile, "Ny rad på slutet\n");
fclose(\$blogfile);

#### Skriv kod som tar hand om informationen från

formuläret och skriver in den i textfilen, uppmärkt med

ā hār:
blog.txt Properties
General Security Details Previous Versions
Object name: W:\TNMK30-2015\blog-file\blog.bt
Group or user names:
Kerveryone
👗 stegu76 (Unix User\stegu76)
&& itn (Unix Group∖itn)
Edit
Permissions for Everyone Allow Deny
Full control
Modify
Read & execute
Read 🗸
Write 🗸
Special permissions
For special permissions or advanced settings, Advanced
click Advanced.
Learn about access control and permissions
OK Cancel Apply

lämpliga HTML-taggar. Taggarna ska läggas till av din kod, bloggaren ska inte behöva skriva in någon HTML-kod i formulärfälten.

Komplettera också din stilmall i CSS (om det behövs) så att posterna blir snyggt presenterade.

# Datummärkning

En blogg bör innehålla information om när inlägg skrivits. **Lägg till kod så att bloggen innehåller en datumstämpel för varje inlägg!** En lämpligt formaterad sträng med dagens datum och aktuellt klockslag kan skapas i PHP med funktionsanropet *\$timestamp = date("Y-m-d H:i:s")*;

# Lösenordsskydd

Dina websidor är helt öppet tillgängliga över Internet, och om man känner till adressen till formuläret kan man posta vad som helst från var som helst i världen. Detta är naturligvis mycket olämpligt ur säkerhetssynpunkt. För att lägga rimliga hinder i vägen för troll och spambottar bör man alltid ha något slags accesskontroll för att lägga till och ändra något via webben. Ett enkelt sätt är att kräva att användaren skriver in ett lösenord i ett extra fält i själva formuläret, och att kolla det lösenordet i PHP. Lösningen har vissa brister, men den fungerar. PHP-koden visas ju inte för användaren, så där kan man skriva lösenordet i klartext. Lägg till ett fält i formuläret för ett lösenord, och kolla om lösenordet är korrekt angivet innan du lägger till en post i bloggen! Om lösenordet är felaktigt bör användaren få ett meddelande om det, till exempel så här:

```
<?php
if ($_POST["password"] != "Hemlis") {
    print("Felaktigt lösenord! Försök igen.");
    }
else {
    // (Lägg till ny post här)
    }
?>
```

# Redovisning

Redovisa din lösning genom att presentera all kod du skrivit. Det blir totalt tre filer: en HTML-fil med formuläret, en HTML-fil med PHP-kod för att spara nya inlägg till fil och en HTML-fil med PHP-kod för att visa de befintliga inläggen. Bloggen ska passas in i strukturen du redan har på din lilla webbplats, och ges en rimligt god grafisk form med CSS.