

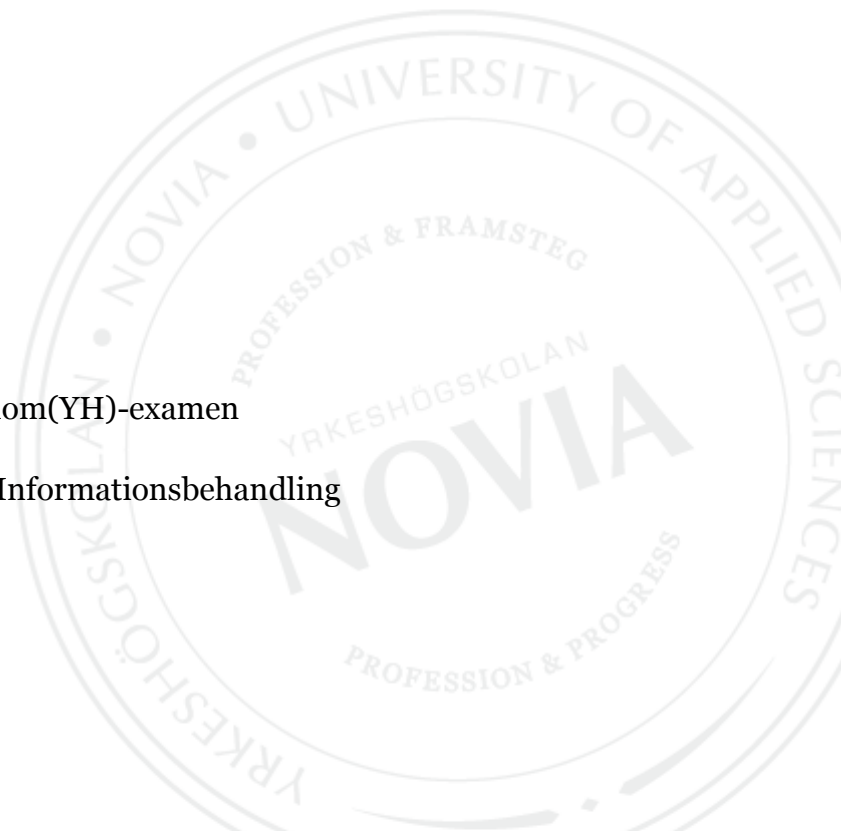
# **Uppdatering av och bättre användarvänlighet för kennelfinby.com**

Sami Kyngäs

Examensarbete för Tradenom(YH)-examen

Utbildningsprogrammet i Informationsbehandling

Raseborg 2014



**EXAMENSARBETE**

Författare: Sami Kyngäs  
Utbildningsprogram och ort: Informationsbehandling Raseborg  
Handledare: Rolf Gammals

**Titel: Uppdatering av och bättre användarvänlighet för kennelfinby.com**

---

Datum 20.11.2014    Sidantal 31    Bilagor 0

---

**Abstrakt**

Detta examensarbete beskriver arbetsprocessen för skapandet av en ny hemsida för Kennel Finby som föder upp hundar av rasen Irländsk setter. Den nya hemsidan ersätter den tidigare hemsidan som användaren upplevde som onödigt svår att uppdatera.

Syftet med hemsidan är att ge information om bl.a. den irländska settern som fågelhund, senaste nytt om kennelns hundar och uppfödningar samt deras resultat från utställningar och prov.

Den största och viktigaste målsättningen med hemsidan var att den skulle bli användarvänlig, dvs bli så enkel som möjligt att använda och uppdatera. I detta skede är texterna endast på finska, men tanken är att i ett senare skede även översätta dem till svenska för att hemsidan även skall nå hundmänniskor i de andra nordiska länderna.

Utifrån detta har hemsidan skapats. De flesta av kundens önskemål kunde även uppfyllas. Arbetsprocessen beskrivs närmare i detta arbete.

---

Språk: Svenska    Nyckelord: PHP, MYSQL, CSS, Webbdesign, CMS

---

## BACHELOR'S THESIS

Author: Sami Kyngäs  
Degree programme: Business Information Technology, Raseborg  
Supervisor: Rolf Gammals

**Title: Updating the website of kennelfinby.com and improving its user-friendliness**

---

Date 20.11.2014

Number of pages 31

---

### **Abstract**

This thesis describes the work process for the creation of a new website for Kennel Finby. This kennel breeds Irish setter dogs. The website is built almost entirely from scratch, as it was unnecessarily difficult to update the kennel's old website.

The purpose of this site is to provide information about Irish setter dogs in particular, to present the latest news about the kennel's dogs, and to show the results from exhibitions and tests.

The largest and most important goal of the website was that it should be user-friendly, which in plain language means that the page should be as simple as possible to use and update. At this stage there is only a Finnish version of the website, but the idea is that also a Swedish version will appear later on, which means an improved contact with other dog people in the Nordic countries.

The result of this thesis work is that the website has been created and most of the customer's requirements have also been met. The entire work process has been described in my thesis.

---

Language: Swedish

Key words: PHP, MYSQL, CSS, Web design, CMS

---

# Innehålls förteckning

<b>1</b>	<b>Inledning.....</b>	<b>1</b>
1.1	Om kunden .....	1
1.2	Den gamla sidans problem .....	2
1.3	Syftet med den nya sidan.....	3
<b>2</b>	<b>Program som använts .....</b>	<b>4</b>
2.1	WinSCP .....	4
2.2	Photoshop .....	5
2.3	Dreamweaver cs6 .....	5
2.4	PHP.....	6
2.5	SQL & MYSQL.....	8
2.6	Xampp .....	8
2.6.1	Apache.....	9
2.7	PhpMyAdmin .....	9
2.8	Webbläsare .....	9
2.9	Jquery.....	10
<b>3</b>	<b>Design av hemsidan.....</b>	<b>10</b>
3.1	Layout på gamla hemsidan .....	10
<b>4</b>	<b>Nya hemsidan.....</b>	<b>12</b>
4.1	Template .....	12
4.2	CSS Style.....	13
4.2.1	Container .....	14
4.2.2	Actionbar .....	15
4.3	Slutgiltiga layouten.....	16
<b>5</b>	<b>Databasen .....</b>	<b>16</b>

<b>6</b>	<b>Funktioner</b>	<b>17</b>
6.1	Index sidan	17
6.1.1	Uppladdning	19
6.1.2	Thumbnails	20
6.1.3	Hämtande av information	21
6.2	Galleriet	22
6.2.1	Thickbox	23
6.3	Require_once	23
6.4	Radera	24
6.5	Gästboken	25
6.6	Administratören	27
6.7	Logga in/ut	29
<b>7</b>	<b>Testning av sidan</b>	<b>29</b>
<b>8</b>	<b>Sammanfattning</b>	<b>30</b>
	<b>Källförteckning</b>	<b>32</b>
	<b>Bildförteckning</b>	<b>33</b>
	<b>Kodförteckning</b>	<b>34</b>

# 1 Inledning

Idén att göra en hemsida som examensarbete kommer från min far som har haft en hemsida för sin kennel i flera år, men vars uppdatering kräver att han skall klara av att koda med html koder vilket är helt obekant för honom. Gamla hemsidan har därför haft sparsamt med uppdateringar om några alls den senaste tiden. En annan sak som fick oss att diskutera en ny hemsida är att servern där gamla hemsidan nu finns på kommer i något skede att avvecklas. Detta var ett ypperligt tillfälle att skapa en ny hemsida. Tanken med detta examensarbete är alltså att skapa en hemsida som fungerar med PHP och MYSQL, och som är så användarvänlig som möjligt för att underlätta för kunden som inte har någon erfarenhet av någon typ av kodspråk från tidigare. Mitt eget mål med denna uppgift är att lära mig mera om hur kodning med PHP fungerar, samt hur jag skall använda databaser. Eftersom jag anser att jag kan för lite om kodning och vill uppdatera mina kunskaper tycker jag att detta arbete där jag på egen hand skall lösa problemen är ett bra val.

## 1.1 Om kunden

Ismo Kyngäs är född och uppvuxen i Posio i norra Finland. Han är utbildad skogsbruksingenjör och skog och mark har alltid haft en stor roll i hans liv i form av egna skogsområden samt jaktmarker. Jakt har han sysslat med i så gott som hela sitt liv och då har han främst jagat skogsfågel. Idag utgör bytet främst ripa, orre och tjäder som han jagar på de egna skogsmarkerna i Posio. Hemma i södra Finland är det fasan som gäller och då främst i skolningssyfte av de egna hundarna eller i samband med att han deltar med hundarna i jaktprov.

Hundar fanns det redan under uppväxttiden i Posio och då var det främst skällande hundar såsom Finsk Spets som han jagade med. I ett senare skede kom apportrande hundar med i bilden och sedan slutligen de stående fågelhundarna. Idag upptar hundarna en stor del av hans fritid. Han satsar aktivt på att skola och delta i jaktprov med dem samt även på uppfödning i mindre skala under kennelnamnet Finby. För tillfället äger han tre stycken irländska settrar varav två är av egen uppfödning och den tredje en tik importerad från Danmark. Han har alltid satsat på kvalitet i uppfödningen genom att välja hundar som är lätt skolade och som fungerar på fågel redan i ett tidigt skede. Detta har resulterat i flera

prisbelönda hundar varav kan nämnas bl.a. Finby Figaro som blev Nylands kenneldistrikts mästare 2014. Eftersom han också föder upp hundar är en fungerande hemsida som är lätt att uppdatera därför av stor betydelse för honom.

## 1.2 Den gamla sidans problem

Den gamla hemsidan har flera problem som bör beaktas och åtgärdas nu samtidigt som en helt ny hemsida skapas. Det absolut största problemet med sidan är att den inte är aktivt uppdaterad. Orsaken till detta är att den inte är skapad på ett sätt som skulle göra den enkel att uppdatera. Om man ville uppdatera sidan så var man tvungen att först koppla upp till själva databasen genom att använda ett program som t.ex. WinSCP. Efter att man har kopplat upp sig så måste man gå igenom alla filer och sedan hitta dem man själv ville ha. Sedan skall man ladda ner allt till sin egen dator. När WinSCP är delat i två delar, hemmatorn och den som man kopplar upp till (se bild 1), har man på vänstra sidan hemmatorn och på högra sidan servern. Detta gör man alltså genom att man drar från ena rutan över till den andra så kopierar det över alla filer.

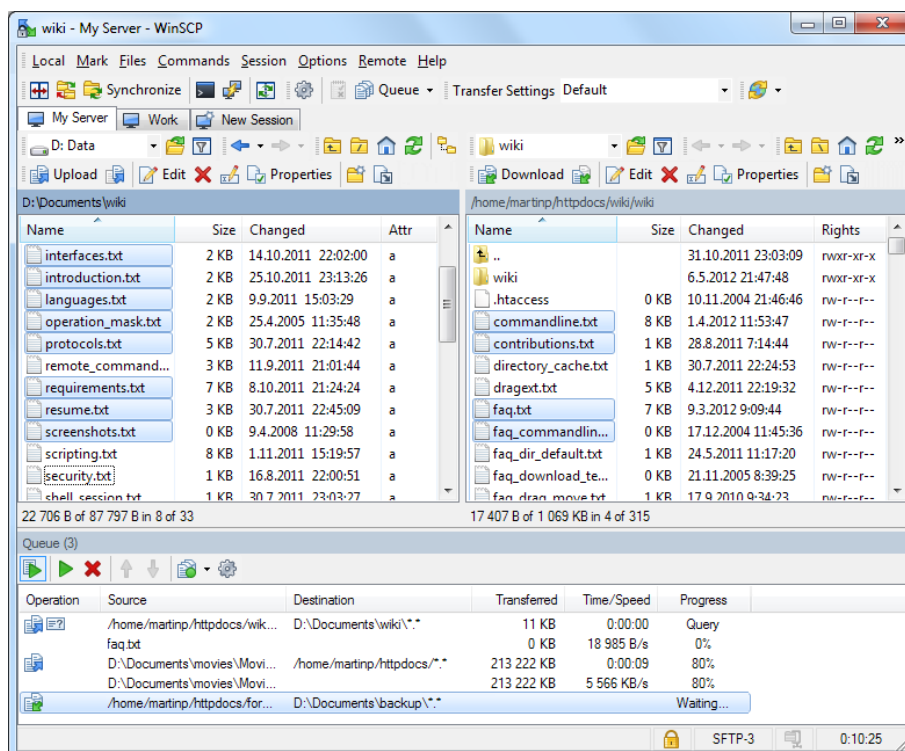


Bild 1. exempel på WinSCP användar gränssnitt.

Efter att man hittat alla filer som man behöver så måste man öppna dem i Dream Weaver eller i ett motsvarande program som gör det möjligt att editera koden. När allt är gjort måste man igen koppla upp sig via WinSCP och ladda tillbaka filerna som man editerat. Eftersom det är så många olika steg samt att man skall använda Dream Weaver så har det varit omöjligt för kunden att sköta detta på egen hand, utan att ta hjälp av någon som förstår sig på kodning mm. Ett annat problem är att sidans utseende har föråldrats. Den är skapad för att synas på skärmar som använder sig av mindre upplösning än de vi idag använder oss av. Detta betyder att ca.40-50% av bilden i webbläsaren är ett tomt utrymme och resten upptas av själva hemsidan.

### **1.3 Syftet med den nya sidan**

Syftet med denna hemsida är att ge information om bl.a. den irländska settern som fågelhund, senaste nytt om kennelns hundar och uppfödningar samt deras resultat från utställningar och prov. All sådan information som kan tänkas intressera eventuella blivande valpköpare, redan ägare av Finby hundar eller personer intresserade av den irländska settern som ras. Detta i form av både text, bild och videon.

Efter diskussion med kunden kom vi fram till följande målsättningar för den nya hemsidan.

- uppdatera hela layouten då den gamla är så liten och fokuserad till vänster, samt radera flera onödiga saker som inte längre är nödvändiga på den nya hemsidan.
- skapa ett galleri som inte kräver något annat än att välja de bilder man vill ha från den egna datorn och sedan med en knapp ladda upp dem till galleriet. Den gamla sidans galleri krävde ändringar i html koden för att bilden skulle synas på sidan.
- möjlighet att lägga till videosnuttar
- gästboken skall finnas kvar på hemsidan, men den skulle byggas om så att den är en del av själva sidan och inte längre länkad till en extern sida.
- uppdatera alla sidorna till dagsläge
- möjlighet att uppdatera hemsidan enkelt

Detta betyder att hemsidan måste byggas helt och hållet om för att den skall uppfylla dessa krav samt vara lätt för kunden att uppdatera på egen hand.

## 2 Program som använts

I nästa stycke går jag igenom alla de program som jag använt för att skapa den nya hemsidan, samt det som jag använt från den gamla sidan.

### 2.1 WinSCP

WinSCP (**Windows Secure CoPy**) är ett program som använder FTP (File Transfer Protocol) vilket gör det möjligt att skicka filer från en maskin till en annan genom att koppla upp med IP adress, portnummer, användarnamn samt ett lösenord (se bild 2) med vilket man kommer åt servrar samt datorer på avstånd. Efter uppkopplingen kan man börja flytta filer från och till maskinen man är uppkopplad till.

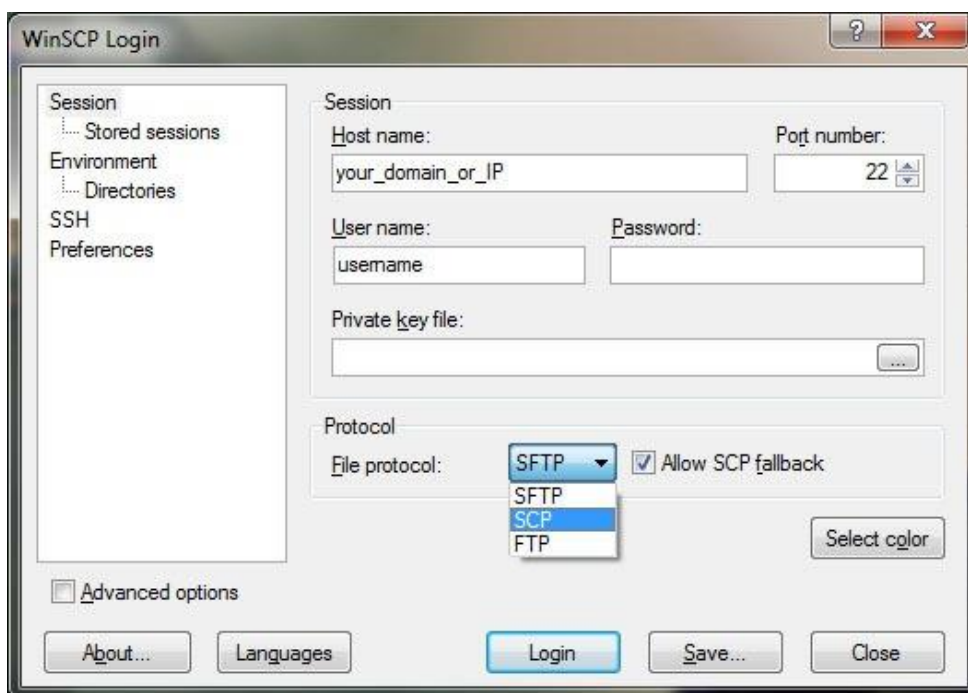


Bild 2. WinSCP uppkopplings fönster

WinSCP var det absolut första programmet som jag använde mig av när jag började med detta arbete.

## 2.2 Photoshop

Första versionen av Photoshop skapades 1990 och baserade sig på programmet Display som skapades av bröderna Thomas och John Knoll. Programmet används av fotografer, grafiker, designers och nybörjare. En stor fördel med Photoshop är att man kan dela upp bilder i lager som gör det möjligt att arbeta på en del av bilden utan att behöva oroa sig över att man ändrar på något annat. Det finns så mångsidigt med verktyg i programmet, hundratals små till stora funktioner, vilket är orsaken till att programmet används av miljoner människor. (Creative bloq, 2005)

Under arbetets gång har jag använt mig av Photoshop bl.a. när jag skapat olika typer av bilder och grafiska delar av hemsidan. Detta inkluderar knappar, bilder, bakgrundsbilden och bannern på sidan.

## 2.3 Dreamweaver cs6

Dreamweaver är gjord för webbdesign samt för utveckling av hemsidor. Det innehåller ett stort urval av funktioner som man behöver när man skapar hemsidor. Programmet är skapat av Macromedia. Dreamweaver blev uppköpt av Adobe systems år 2005 och sedan dess har det hett Adobe Dreamweaver. En av fördelarna med att använda detta program är att du har möjlighet att se förändringarna på hemsidan i realtid när du arbetar.

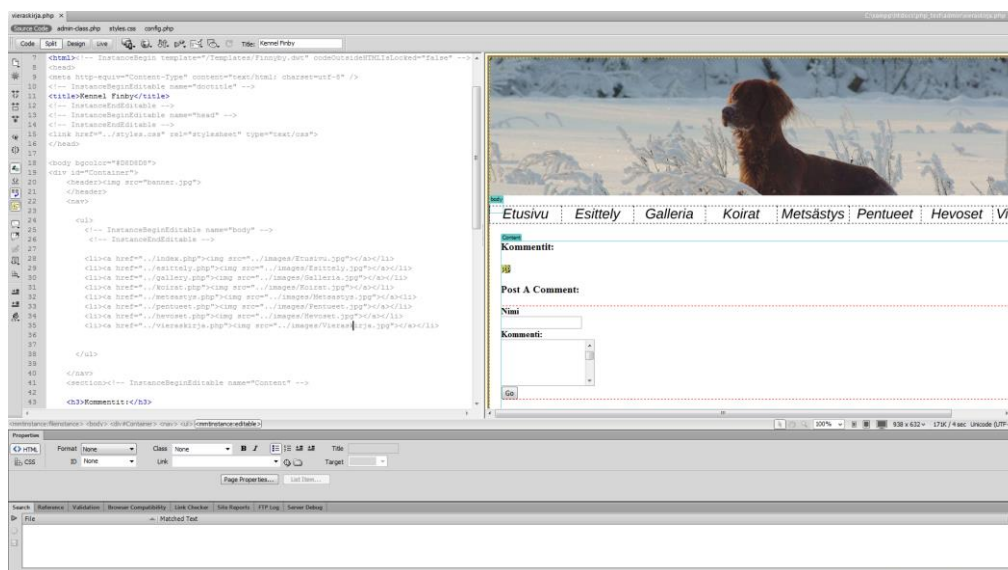


Bild 3. Exempel på Dream Weaver arbetsfönster

## 2.4 PHP

PHP (Hypertext preprocessor) är ett öppet källspråk som är känt för att vara mycket användarvänligt för nya användare. Det gör det lätt för sådana personer att komma i gång som inte vanligtvis kodar mycket samtidigt som det erbjuder många avancerade funktioner för de som arbetar med programmering. Skillnaden mellan PHP och HTML (Hypertext Markup Language) är att allt som är HTML körs i själva klienten medan det som är PHP körs på serversidan. Detta betyder att när PHP kör några flera punkter på serversidan så hoppar html över det och kör det i webbläsaren (se bild 4). I detta steg hämtar den upp filen man begärt och skapar en html kod från det. Ifall det skulle köras som en html fil istället för PHP så skulle den hoppa över steg 4 och 5 (se bild 4), och istället köra steg 6 direkt.

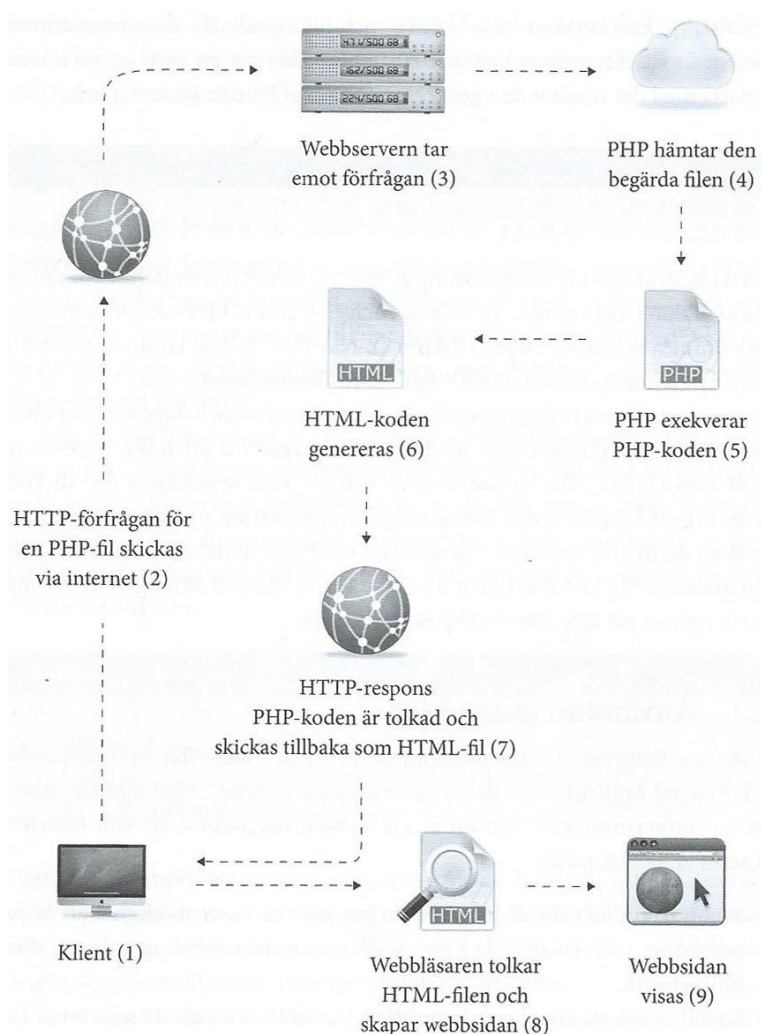


Bild 4. Händelseförloppet för PHP (Webbutveckling med PHP och MYSQL, 2012, s.17)

En av de tydligaste fördelarna är att du kan tvinga programmet att köra allt som HTML i ett skript. Detta möjliggör att den inte visar koden som körs på serverns sida utan bara det som kommer från servern fram till hemsidan. (*Webbutveckling med PHP och MYSQL, 2012, s.16-17*)

Andra nämnvärda funktioner med PHP är att du kan begränsa vad kunden kan eller inte kan göra på hemsidan. Om man inte är inloggad som administrator kommer man inte att kunna komma åt de sidor som använder sig av administrator funktionen, t.ex. de funktioner som laddar upp ny information till servern eller editerar vad som redan finns på databasen.

Programmet har också möjlighet att samla in all data från en viss tabell på en databas för att sedan visa upp det på en viss sida i ett visst format.

Fast PHP är mest fokuserad på serverskript, som att skicka eller läsa/ta emot från databasen, kan man göra mycket mera med den, t.ex. skriva skrivbordsprogram och kommande prompt script. Man kan säga att det finns nästan obegränsade möjligheter med PHP. Att skapa dynamiska webbplatser och göra uppgifter som brukade vara långtråkigt arbete, kan man nu förenkla genom att använda PHP och på det här sättet få helautomatiserade skript som hanterar allt på databasen genom ett klick på en knapp (kod exempel 1).

*Kod exempel 1. PHP skript som visar "Detta är en PHP-funktion!" när den körs.*

```
<!DOCTYPE HTML>
<html>
<body>
<? PHP
ECHO "Detta är en PHP-funktion!";
?>
</body>
</html>
```

## 2.5 SQL & MYSQL

SQL (Structured Query Language) används vid databas programmering och är skapat av Donald D Camberlin och Raymond F. Boyce på IBM i början på 1970 talet.

(msdn.microsoft)

Programmet fungerar med förfrågningar så det kan både skapa, infoga, radera samt editera databaser. Man behöver heller inte oroa sig över små och stora bokstäver när man använder sig av SQL.

Denna kod skall söka från tabellen som heter pentuea samt ordna alla inlägg av namnen i fallande ordning. Eftersom det gäller hundkullar är det praktiskt att ha dem i alfabetisk ordning (kod exempel 2).

*Kod exempel 1. Query skript som söker från pentuea tabellen och sorterar enligt namn.*

```
$data = mysql_query("SELECT * FROM pentuea order by `nimi` DESC ")  
or die(mysql_error());
```

Om man använder samma typ av kod för en indexsida där man vill byta nimi till ID, skulle det då börja sorteringen från det största ID-numret till det lägsta. Ju nyare inlägget i databasen är desto större skulle ID numret bli eftersom den alltid sätter plus ett till ID numret. Ju äldre inlägg desto mindre ID nummer skulle det ha, vilket gör att det kommer längre ner i listan när det hämtas från tabellen i servern till hemsidan.

## 2.6 Xampp

XAMPP (Cross-Platform (X) Apache (A) MySQL (M) PHP (P) och pearl (P)), är ett gratis öppenällkodsverktyg som används för utveckling av hemsidor. Programmet används för att skapa virtuella servrar på den egna hemmaskinen, vilket möjliggör att man inte behöver testa hemsidans funktioner på en riktig server. Detta program underlättar mycket då man kan testköra alla funktioner som olika PHP koder med databasen samt lösa alla problem som kan uppstå. På detta sätt kan man frångå större problem och behöver inte flytta filer mellan hemmaskin och servern hela tiden.

## 2.6.1 Apache

Apache är webbservern som jag använde mig av då jag testade hemsidan under tiden som jag arbetade med de olika PHP funktionerna. Jag valde att använda denna eftersom den var inbakad i själva Xampp-programmet när jag började använda det.

## 2.7 PhpMyAdmin

PhpMyAdmin är ett öppen-källkodsverktyg som tillåter dig att hantera administrationen av din My SQL-databas i ett användargränssnitt i din egen webbläsare. Då programmet är skrivet i PHP kan det enkelt hjälpa dig att skapa databaser, ta bort eller redigera databaser, bygga och redigera tabeller samt hantera fält i databasen.

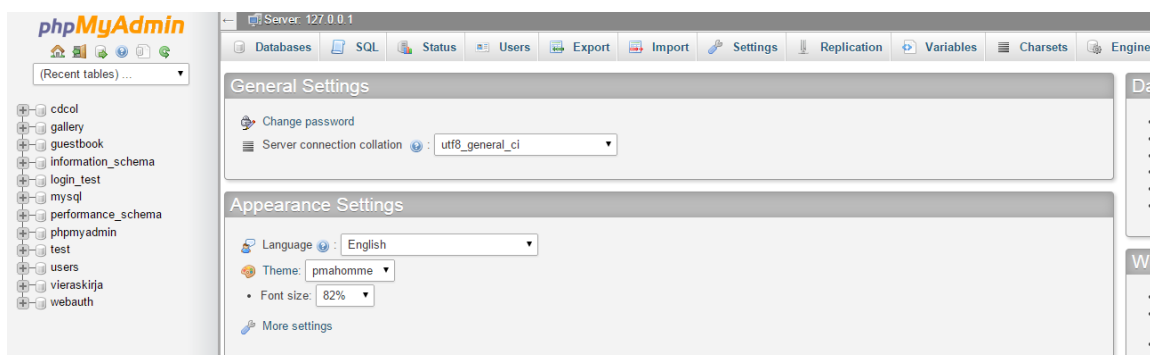


Bild 5. Denna bild visar ett exempel på hur phpMyAdmin ser ut i webbläsaren.

Programmet fungerar väl i linje med Xampp när det gäller att hantera och skapa testdatabaser då du håller på att utveckla. Du kan köra alla typer av PHP funktioner utan att behöva skaffa en riktig server där du testkör dem. (phpMyAdmin)

## 2.8 Webbläsare

Då jag arbetade med hemsidan kontrollerade jag regelbundet att allting såg ut som det borde. Till detta använde jag de tre vanligaste webbläsarna Firefox, Google Chrome och Internet Explorer. Dessa fanns i Dream Weavers förhandsgranskningsfunktion, vilket gjorde det möjligt att ladda fram hemsidan i den webbläsare som man ville pröva den på.

## 2.9 JQuery

”jQuery är världens mest populära JavaScript-bibliotek. Det är designat bl.a. för att göra det enklare att anropa och modifiera element på hemsidan (divar, bilder, länkar etc.), hantera händelser (t.ex. musklick), CSS-modifikationer samt skapa effekter och animationer. jQuery stöder också plugin. Det finns många smidiga plugin vilka gratis går att ladda ner och använda.” (jQuery-guiden (u.å.))

## 3 Design av hemsidan

Som följande kommer jag att gå djupare in på den gamla sidans layout och dess problem. Jag kommer att berätta stegvis hur den nya layouten är uppbyggd samt vad som är gjort för att förbättra och få den mera lätt att uppdatera.

### 3.1 Layout på gamla hemsidan

När det gäller utformningen av den gamla hemsidan var jag tvungen att förkasta det mesta. Jag tog säkerhetskopior på redan uppladdade bilder och information. Själva sidorna var det ingen idé att spara för allt som där fanns var skrivet i html format. Själv planerade jag att laga om den till PHP.

Jag hade först tänkt gå den lätta vägen och enbart börja arbeta om den gamla hemsidan. Detta skulle ha betytt omeditering av själva utseendet på sidan samt uppbyggande av alla de nya funktioner som vi diskuterade med kunden. På detta sätt hade jag räknat med att spara tid, men det visade sig bli ett större arbete än vad jag från första början planerat.

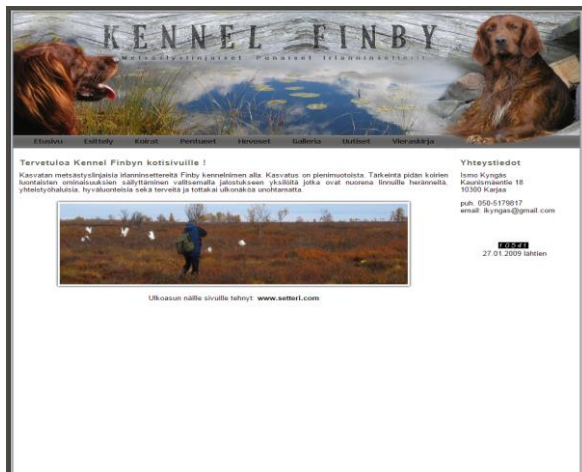


Bild 6. Exempel på hur gamla sidan såg ut.

Allt eftersom jag gick igenom den gamla sidan för att få en bild av hur jag lämpligast skulle gå till väga, blev det klart att den inte var planerad på ett sätt som skulle göra det möjligt att uppdatera den. Då den ursprungliga hemsidan skapades så gjordes den som en statisk bild som fungerade som sidans layout. Bredden på den gamla sidan var för liten när bilden var 960 pixel bred. Om jag önskade ändra på den ursprungliga sidan så måste jag öppna upp allt i Photoshop, klippa ut onödiga saker samt försöka öka på bredden. Om hemsidan skulle ha varit skapad på en template från första början så skulle det ha varit möjligt att ändra på koden i små delar och sedan byta ut bilder och knappar.

Nu då hemsidan är skapad som en bild, är den uppbyggd så att hemsidans knappar alla är genomskinliga rutor, placerade under själva bilden (Se bild 7). Knapparna reagerar när muspekaren förs utanpå dem.

Om jag skulle ha editerat om meny balken skulle jag också ha behövt ändra varje enskild sida för hand så att de skulle se lika ut som det som jag ändrade.



Bild 7. Gamla sidans menybalk

Före jag började arbetet med den nya hemsidans layout tog jag reda på vad som idag är den vanligaste upplösningen för bildskärmar. Från loggfiler på w3schools.com hemsida har de kommit fram till att 99 % av alla besökare har en upplösning av 1024x768 eller större. Från bilden nedan (bild 8) kan man utläsa att trenden är att övergå till skärmar med större upplösning. Jag kan alltså om jag så önskar ganska tryggt göra en sida med en bredd på 1366px eller större.

Date	Other high	1920x1080	1366x768	1280x1024	1280x800	1024x768	800x600	Lower
January 2014	34%	13%	31%	8%	7%	6%	0.5%	0.5%
January 2013	36%	11%	25%	10%	8%	9%	0.5%	0.5%
January 2012	35%	8%	19%	12%	11%	13%	1%	1%

Bild 8. Information på skärmarnas upplösning samlad av w3schools.(w3schools.com)

## 4 Nya hemsidan

Som följande kommer jag att gå in på de olika steg som krävdes då jag skapade den nya hemsidans layout.

### 4.1 Template

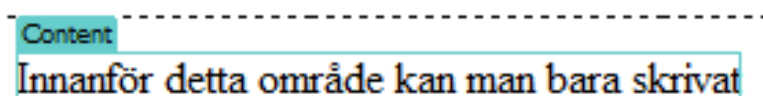
Eftersom jag valde att göra om hemsidan helt från början beslöt jag att göra en template fil för hemsidans hela strukturen. Den ursprungliga hemsidan var gjord av en bild. Om jag inte skulle ha valt template så var mitt andra alternativ att skapa en kod för sidans struktur som sedan skulle kopieras i efterskott till varje sida jag skapat. Detta skulle ha fungerat men det skulle ha blivit mera arbetsdrygt att uppdatera varje flik då layouten ändras.

Största fördelen med att använda template är att jag kan låsa bort vissa delar av sidans kod och göra vissa områden så att det går att editera och skriva på dem (kod exempel 3). Om man i ett senare skede önskar skapa nya sidor till hemsidan är det enkelt att använda skapa new page from template i Dream Weaver. Detta laddar upp en färdigt gjord template med allt färdigt. Det enda man sedan behöver göra är att skapa innehållet och länka det på rätt plats

*Kod exempel. 2 CSS kod på område som kan editeras av användaren.*

```
<!-- TemplateBeginEditable name="Content" →
Innanför detta område kan man bara skriva
<!-- TemplateEndEditable --></section>
```

Det område man kan editera på hemsidan ser ut enligt bilden nedan (se bild 9). Väljer man något utanför rutan går det inte att ändra på det om man inte går direkt till template filen.



*Bild 9. Området innanför den blåa rutan kan editeras, allt utanför är låst.*

Då allting på själva layouten är låst utanför templatens kan kunden inte gå in på en viss sida och ändra färgen eller byta bannern på sidan. Varje gång kunden önskar göra detta måste kunden komma åt själva template filen. På detta sätt säkrar jag att alla sidor alltid ser likadana ut. Inne på själva template filen går det att ändra hemsidans layout, banner eller sidfot bild. Görs det här ändrar varje sida på hemsidan, Det är ingen skillnad hur många sidor man har för alla sidor använder samma modellfil för strukturen.

## 4.2 CSS Style

CSS (Cascading Style Sheets) är en kodningsstil gjord för att definiera allt på hemsidans layout. Det går att sätta b.l.a. följande regler för sidans layout. Färg, höjd, bredd, bakgrundsbild, marginaler, font och mycket annat. Fördelen med att använda CSS är att den sparar all information om hemsidans struktur i ett skilt dokument vilket gör det mycket lättare och tydligare att veta vad som är information på hemsida och vad som är själva strukturen av sidan då man ser på koden. Med ett CSS dokument kan man styra hur många sidor som helst. Det enda som krävs vid skapandet av nya sidor i efterskott är att varje sida kopplas till detta CSS dokument.

Innan jag börja arbeta på CSS filen skapade jag i Photoshop en snabb layout template för att få en uppfattning om hur jag vill att hemsidan skall se ut (Se bild 10). Kunden önskade hålla kvar den gamla hemsidans layout då det var lätt att navigera mellan de olika flikarna. I Photoshop skulle det ha funnits färdiga versioner att använda men då jag önskade få en uppfattning om hur CSS filen är uppbyggd gjorde jag allting från början.



Header	1024x180
Action Bar	1024x50
Content Zone	1024x
Footer	1024x60

Efter att jag skapat en version på hur jag önskar att hemsidan skall se ut, var det lättare att börja arbeta med själva CSS filen. Eftersom det är en riktgivande version så är inget fastställt slutgiltigt. Saker och ting kan ändra ännu enligt behov. Efter övervägande beslöt jag dock använda det säkrare alternativet angående sidans layout och skapade därför sidan med 1024px i bredd. Nu skall den synas bra på de flesta data skärmar.

### 4.2.1 Container

Det första steget blir att skapa en container vilken kommer att fungera som det område som sidans funktioner visas på (kod exempel 4). Då jag har exempel layouten gjord gör jag den 1024px bred som första version och bakgrunden vit till att börja med. Då syns texten bra.

*Kod exempel 3. CSS hela hemsidan kommer att finnas innanför container området.*

```
#Container{  
width: 1024px;  
margin: 0 auto;  
background-color: #FFF;  
}
```

Efter att jag skapat en container gör jag sedan resten av CSS sidans kod. Nästa steg blir att sätta ut sidhuvud, navigeringsbalk och sidfot för sidan. Åt body elementet ger jag ingen maximi höjd då den kommer att sträcka sig vid behov när man fyller sidan med information. Enda begränsningarna jag gjorde var att den inte skall börja riktigt fast i vänstra kanten och att den skall sluta lite före högra kanten av sidan (kod exempel 5). Då sitter inget fast i hemsidans ram.

*Kod exempel 4. CSS de enskilda sidornas funktioner kommer att fungera innanför detta område.*

```
section{  
float: left;  
padding-left: 20px;  
width: 984px;  
}
```

Resten bör ha fasta former så att de inte ändrar sig. Själva sidhuvudet och sidfoten görs med nästan samma identiska kod som containers. Den enda skillnaden är att de får en höjd som bestämmer hur stort område det totalt får ta. Enda skillnaden med sidfoten är att jag skrev kommando `bottom: absolute;` vilket gör att den alltid är på botten av hemsidan.

#### 4.2.2 Actionbar

Actionbar är den del av sidan som kommer att användas mest med kod av hela CSS dokumentet. Här sätter man ut själva höjden och bredden på balken och hur stora varje knapp skall vara på för att det skall se bra ut.

När jag hade klart att jag kommer att skapa åtta stycken flikar på balken, delade jag 1024 med åtta för att få reda på hur stor varje knapp maximalt får vara för att de inte skall visas fel på webbläsaren (kod exempel 6). Hade jag valt mindre knappar och mellanrum så skulle jag ha kunnat använda mig av padding (tomt utrymme som man kan sätta mellan bilder så att de inte sitter fast i varandra) som skulle ha satt extra bredd på dem för att de skulle spridas ut bättre.

*Kod exempel 5. Koden visar knapparna på actionbaren.*

```
nav ul li a:link, nav ul li a:visited{
display: block;
height: 30px;
width: 128px;
line-height: 30px;
text-align: center;
padding:0px;
}
```

### 4.3 Slutgiltiga layouten

Den slutliga layouten av hemsidan blev ett format som skall vara så lätt som möjligt att navigera och inte innehålla något onödigt (Se bild 11).

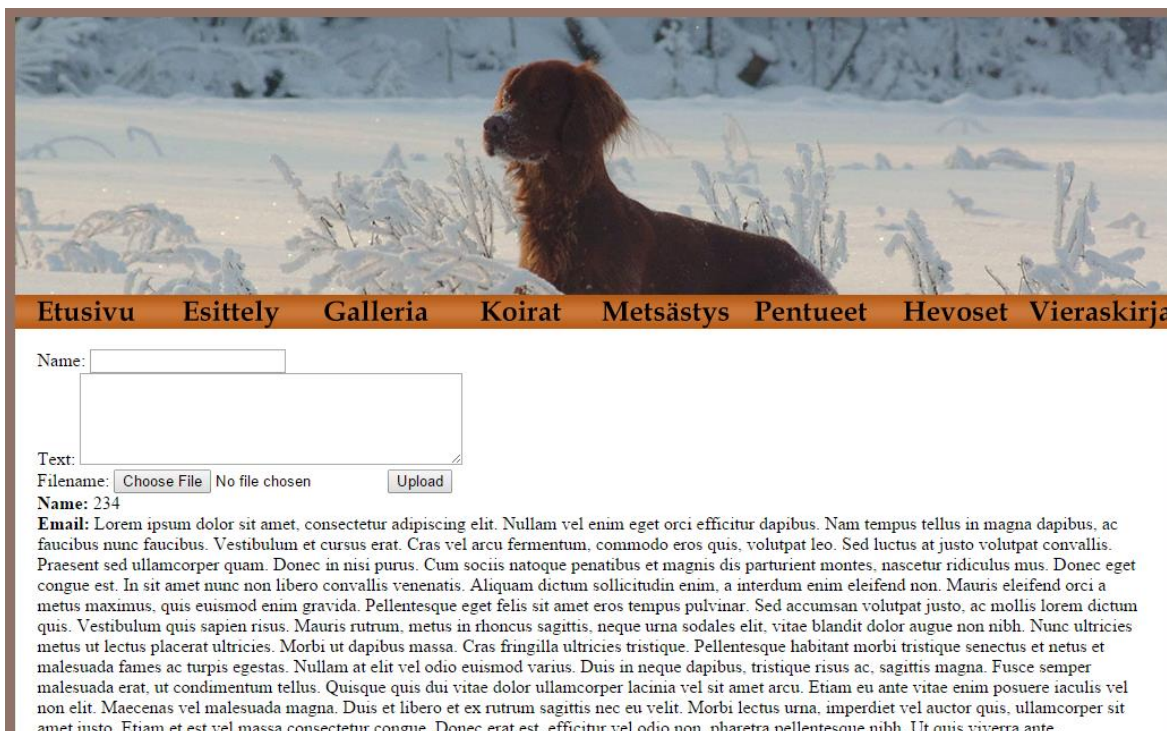


Bild 11. Layout på den nya hemsidan.

## 5 Databasen

Databaserna är en samling av information som är lagade så att de skall vara lätta att koppla upp till samt göra det möjligt att söka och editera information från dem utan större problem. En databas finns på en server som klarar av att upprätthålla flera databaser. För att göra saker lättare med flerspråkiga sidor kan man köra med olika databaser så att information är lätt att hitta samt editera.

Databaserna är sedan splittrade i flera mindre delar som kallas tabeller. Tabellerna är sedan delade i rader och kolumner vilket gör det möjligt att spara flera olika typer av information. I en rad kan man spara flera fält med information som t.ex. i gästboken där jag sparade namn, kommentar samt tiden då inlägget skrevs (se bild 12).

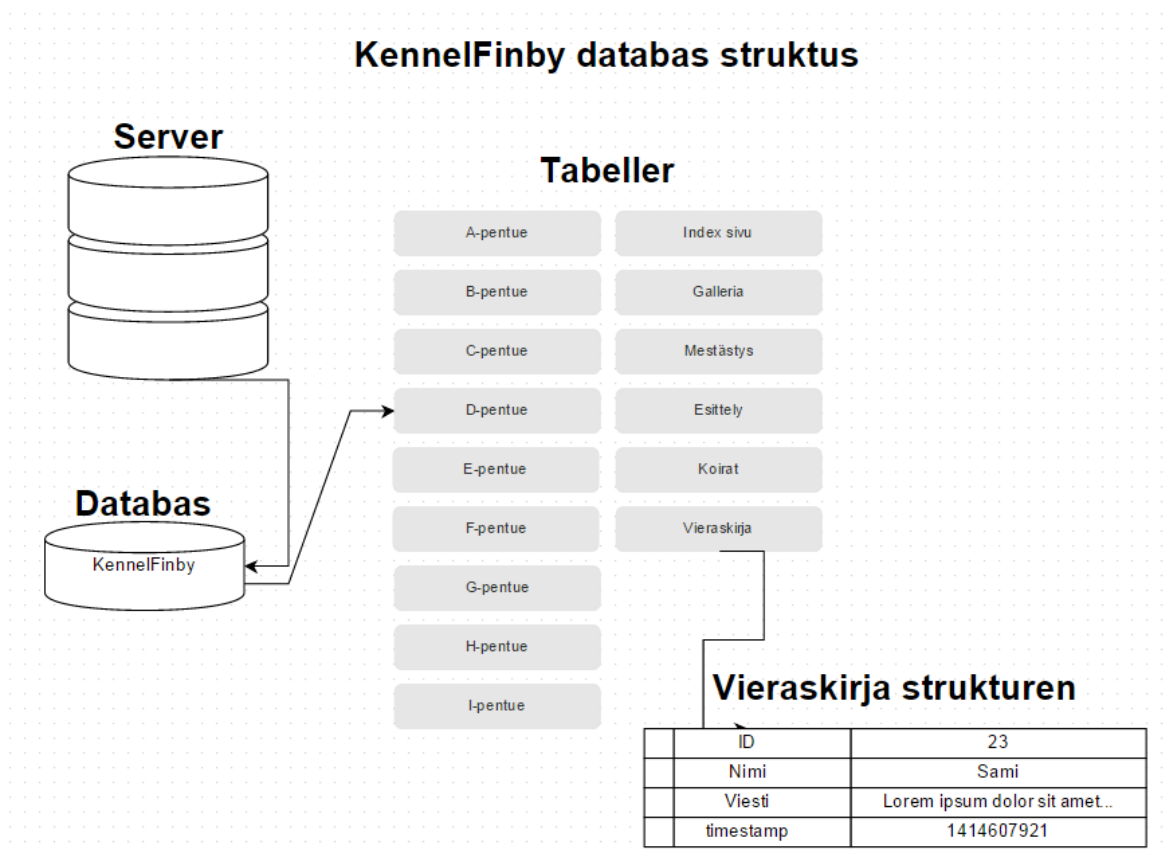


Bild 12. KennelFinby databas struktur

När jag har skapat en bild på strukturen av databasen övergår jag till databasen och börjar skapa varje tabell skilt som jag kommer att behöva. Till först kommer de inte att innehålla information. Efterhand då jag arbetar på sidans olika delar kommer de att fyllas.

## 6 Funktioner

Under funktionsdelen kommer jag att gå igenom hemsidans olika aspekter som jag arbetade på och hur jag gick skapade allt på index sidan, galleriet, gästboken och de andra delarna.

### 6.1 Index sidan

Index sidan är den absolut viktigaste sidan av detta arbete för det är den besökaren ser först då han besöker hemsidan Målet med index sidan är att den skall vara så lätt som möjligt att uppdatera samt att man sätter max en bild per inlägg. Denna bild skall vara i thumbnail

format så att när man klickar på den en gång öppnas bilden i en flik. Resten av bilderna skall sedan finnas i hemsidans galleri. Jag gjorde detta val för att få index sidan att ladda upp så snabbt som möjligt (utan att behöva ladda stora mängder med information). Koden skall på egen hand skapa en thumbnail fil från den ursprungliga bilden utan att man skall behöva skapa två bilder.

Arbetets första steg var att sätta upp själva index sidan i databasen, så att det skulle vara möjligt att ladda upp information. När allt som man behöver finns i databasen är det lättare att börja arbeta på själva koden. Då allt är färdigt upplaggt vet man exakt vad man skall ha för namn på allt i själva PHP koden och behöver då inte skapa det i efterskott.

Jag visste att index sidan behöver några fält för att fungera så till först gjorde jag titeln, texten och bilden. Date\_added behövdes för att man skall kunna se när det senaste inlägget är skapat. ID gör att de sorteras i rätt ordning. I bilden (bild 13) kan man se att jag valt att ge titeln samt bild namnen en möjlighet att använda ganska många tecken, 200 och 250. Detta är kanske lite extremt mycket för ett bild namn samt för titel. Jag gav teksten som longtext då jag inte vill sätta en gräns för hur många tecken kunden kan använda när han skriver inlägg på hemsidan.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 <b>id</b>	int(30)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2 <b>photo_name</b>	varchar(200)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	3 <b>otsikko</b>	varchar(250)	latin1_swedish_ci		Yes	NULL	
<input type="checkbox"/>	4 <b>teksti</b>	longtext	latin1_swedish_ci		Yes	NULL	
<input type="checkbox"/>	5 <b>date_added</b>	datetime			No	None	

Bild 13. Strukturen av index sidan i databasen.

När databasen är gjord och färdig att ta emot all information och alla filer är nästa steg att skapa skripten som skickar allt till databasen. Den här delen av arbete har mest med PHP kod eftersom den skickar all text samt skall klara av att skapa två versioner av bilden. En i original storlek och en i mindre version för att spara utrymme på index sidan. På grund av detta behöver den inte ladda stora mängder för att visa sidan. När man trycker på den mindre bilden för den över till den ursprungliga versionen. Därför behöver den skapa och

skicka bilderna till två olika mappar som finns under namnen thumbnails för den mindre bilden och uploads för den andra.

### 6.1.1 Uppladdning

När det till databasen laddas upp bilder samt text, behöver jag några fält otsikko, teksti samt en sökfunktion vilken gör det möjligt att söka upp bilder från en maskin (kod exempel 7). När man laddar upp skall varje fält fyllas i och därefter väljer man bilden man vill ladda upp.

*Kod exempel. 6. PHP skripten för att skapa fälten som används vid uppladdning av information.*

```
<form enctype="multipart/form-data" action="indexadd.php"
method="POST">
  Name: <input type="text" name="name" width="200"><br>
  Text: <textarea name="email" cols="45" rows="5"> </textarea><br
/>
<label for="file">Filename:</label>
<input type="file" name="file_upload" id="upload_file" />
<input type="submit" name="submit" value="Upload" />
</form>
```

De fält man laddar upp med skiljer sig inte från de andra sidorna på hemsidan.

För att undvika att man inte skall kunna ladda upp hurdana filer som helst, finns det inbyggt i koden att man bara kan ladda upp bilder som är i .jpg och .jpeg format (kod exempel 8). Om man inte sätter en restriktion så är det möjligt att ladda upp vad som helst.

*Kod exempel. 7. PHP skript som sätter gräns på vilken filtyp som är tillåten.*

```
$SallittuMuoto= array("jpg", "jpeg");
...
if ((($_FILES["file_upload"]["type"] == "image/jpg")||
($_FILES["file_upload"]["type"] == "image/jpeg"))&&
in_array($extension, $SallittuMuoto))
```

Vad den också skall kontrollera är att det är i rätt format samt att namnet inte är upptaget i databasen (kod exempel 9). Detta gör det för att den inte skall skriva över bilder som tidigare har laddats upp i databasen. Ifall den skulle skriva över, betyder det att det äldre inlägget skulle ladda in den nya bilden med samma namn och förstöra den gamla bilden. För att detta skall fungera kör den ett skript som kontrollerar om namnet redan är upptaget.

*Kod exempel 8. PHP skripten ser ifall namnet är redan upptaget i databasen.*

```
if (file_exists("../admin/uploads/" . $photoname))
    {
die( '<div class="error">Anteeksi <b>'. $photoname . '</b> löytyy
jo tietokannassa</div>');
    }
```

Sista funktionen den kör är att skapa thumbnail filen. För detta söker den bilden från uploadsmappen och för över den till thumbnailsmappen (kod exempel 10). Nu ger den det största ID talet och sätter det bakom namnet av thumbnail bilden.

*Kod exempel 9. PHP skripten söker original filen och kör make\_thumb funktion som skapar thumbnail.*

```
$source = "../admin/uploads/" . $photoname;
$destination = "../admin/thumbnails/thumb_" . $maxId . $ext ."";
$width = "282";
make_thumb($source,$destination,$width);
```

### 6.1.2 Thumbnails

När det skapas två versioner av bilden och den mindre bilden används på själva index sidan, ger detta möjligt för hemsidan att ladda upp mycket snabbare än om det skulle vara en ihoptryckt miniatyr bild. Använder man en ihoptryckt bild, laddar den upp ursprungliga filen vilken då är i mindre format medan storleken inte skulle minska. Detta betyder att sidan skulle ladda upp betydligt långsammare.

Som jag nämnde i punkt 6.1.1 skickar funktionen bilden till två olika mappar på servern. En som innehåller original bilden och en thumbnail versionen. Dessa två behövs när information hämtas. Eftersom programmet skapar bilden på egen hand, är den inte av samma kvalitet som om den vore gjord i Photoshop eller något liknande program. Men då uppgiften är att underlätta arbetet för användaren, får kvaliteten vara lite sämre. För thumbnail funktionen använder jag ett färdigt skript som är skapat av David Walsh (kod exempel 11), eftersom jag inte själv klarade av att skapa något som skulle fungera på detta sätt.

*Kod exempel 10. PHP Skript som skapar thumbnail (David Walsh kod)*

```
function make_thumb($src, $dest, $desired_width) {
    $source_image = imagecreatefromjpeg($src);
    $width = imagesx($source_image);
    $height = imagesy($source_image);
    $desired_height = floor($height * ($desired_width / $width));
    $virtual_image = imagecreatetruecolor($desired_width,
    $desired_height);
        imagecopyresampled($virtual_image, $source_image, 0, 0,
    0, 0, $desired_width, $desired_height, $width, $height);
        imagejpeg($virtual_image, $dest,100);
}
```

### 6.1.3 Hämtande av information

När sidan hämtar informationen från databasen, söker den titeln, texten och bilden. Bildens information hämtas från två fält i databasen. Den kombinerar thumbnail filen med ID numret på inlägget och länkar sedan thumbnail bilden till den ursprungliga (kod exempel 12).

*Kod exempel: 11 PHP skript som hämtar bildens information från databasen och visar den på index sidan.*

```
$file = $info['kuva_nimi'];
echo '<div id="container">
<div id="thumbnail"><a href="uploads/'. $file .' title="'. $file .'">
</a></div>
<div id="info"><strong>'. $info['titel'] .'</div>
</div>';
```

I kod exemplet tar den från thumbnail som tillhör inlägget genom att söka den med kod delen "thumb\_'.info['id']", Den söker fram thumbnail bilden enligt inläggets ID nummer. Om inlägget har nummer 13 så hämtar den thumb\_13.jpeg. Varje thumbnail börjar med "thumb\_" och sedan ett nummer som kopplar ihop med ID talet. Länken till den ursprungliga bilden tas från databasen genom att kombinera ID numret och kuva\_nimi. De andra delarna som hämtas från databasen är lika som för de andra sidorna, titeln, datum och texten som tillhör dem.

## 6.2 Galleriet

Hemsidans galleri fungerar som ett utrymme där kunden kan ladda upp alla bilder han inte kan lägga på index sidan. I hemsidans första version finns det bara ett galleri. Tanken är dock att i ett senare skede laga flera olika gallerier, så att kunden har möjligt att välja olika kategorier på bilderna. Valet att skapa bara ett galleri i början gjordes för att spara tid med utformningen av sidan. Målet är att få den så snabbt som möjligt i bruk. I ett senare skede kan man utveckla den mera enligt behov.

Galleriet använder sig av samma typ av thumbnail skript som index sidan. Allt som finns är själva bilderna samt deras namn under dem. Eftersom jag använder thickbox, kommer den att besluta hur många bilder det kommer att finnas per rad (Se bild 14). Den ser hur många som ryms och ifall det inte ryms flera så påbörjar den en ny rad. Bilderna kommer att vara tre stycken per rad.

# Kennel Finnby Galleria

23

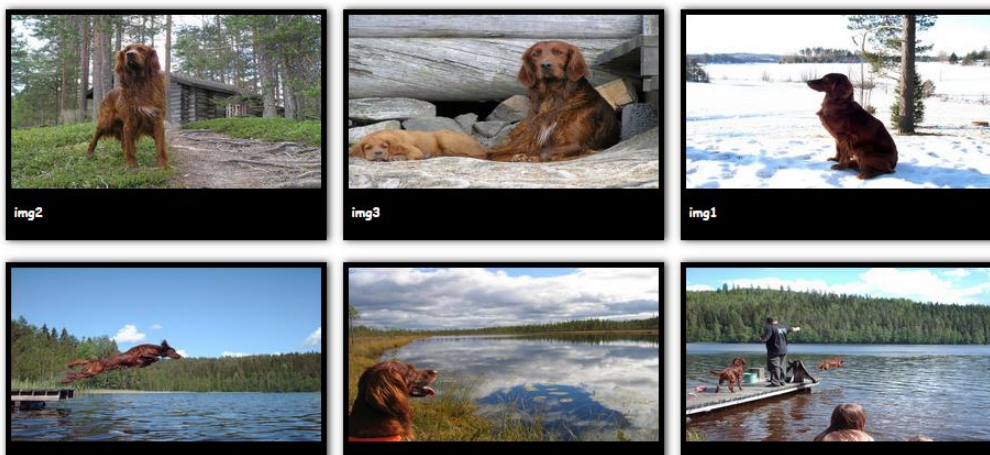


Bild 14. Exempel på galleriets struktur.

För att thickbox funtkionen skall fungera, skall man koda in på sidan att den skall använda sig av thickbox css och js(java script)(kod exempel 13)

Kod exempel 12. Länkar thckbox.css till galleri sidan.

```
<script type="text/javascript" src="js/thickbox.js"></script>  
<link rel="stylesheet" href="css/thickbox.css" type="text/css"  
media="screen" />
```

## 6.2.1 Thickbox

För att få rätt utseende på galleriet, har jag använt jag mig av thickbox vilket är en lättare version av JQuery, Detta gör det möjligt att skapa bl.a. ramar runt bilderna i ett galleri. Thickbox fungerar som så att när bilden har öppnats, är den alltid centrerad oberoende om man rör sig uppåt eller nedåt på sidan. Bilden är alltid i mitten. Samma funktion fungerar då man zoomar in och ut. (Lindley, 2007)

## 6.3 Require\_once

För att underlätta själva kodandet av hemsidan så skapade jag en config.php fil som används på alla sidor som har något med databasen att göra. Den innehåller inte mycket koder men den har allt som gör det möjligt att koppla upp till databasen. Den behövs varje gång man vill ladda upp eller hämta information från databasen.

Då jag använder mig av `require_once` kontrollerar den varje gång att den har kopplat upp till databasen och ifall den inte har gjort det kommer den att köra filen innan den gör något annat. Filen innehåller den nödvändiga information som krävs för att koppla upp till databasen. Användarnamn, lösenord och databas namnen och hosten (kod exempel 14).

*Kod exempel: 13. PHP alla nödvändiga fälten för att koppla till databasen.*

```
<?PHP
$host = "localhost";
$username = "root";
$password = "pass";
$database = "kennelfinby";
mysql_connect($host,$username,$password);
mysql_select_db($database);
?>
```

## 6.4 Radera

Raderingsfunktion behövs för att underlätta arbetet ifall man gör något fel eller vill ta bort någonting. Om den inte skulle finnas på administrationssidan skulle man behöva gå in på databasen och därifrån sedan söka fram det inlägg man vill radera från databasen. Med denna funktion klarar man av att snabbt radera flera inlägg genom att enbart trycka på en länk på sidan. Det är uppbyggt så att varje rad som hämtas från databasen kommer att ha en länk till `delete.php`. När man trycker på länken kommer den automatiskt att köra koden. (Kod Exempel 15). Koden kontrollerar vilket ID nummer den är kopplad till och sedan raderas bara det inlägget.

*Kod exempel: 14 PHP skript som raderar inlägg baserat på ID nummer.*

```
<?php
require_once 'config.php';
mysql_query("DELETE FROM index WHERE ID =$_GET[id]")or
die(mysql_error());
echo "Päivitys on poistettu!";
header('Location: index.php');
?>
```

## 6.5 Gästboken

Eftersom kunden önskade behålla gästboken kvar beslöt jag att skapa den på nytt i ett PHP format, istället för den gamla gästboken som var länkad till en utomstående hemsida. Jag upplevde det som konstigt att gamla hemsidans gästbok förflyttade till en helt annan webbsida.

Eftersom gästboken är en enkel funktion på nya hemsidan krävdes det inte mycket arbete för att få den att fungera. Först skapades själva databasen som kräver enbart fyra rader. Här kommer inte att sparas stora mängder information varför den enbart består av id, nimi, viesti och timestamp (se bild 15).

För att få alla kommentarer i rätt ordning använder jag timestamp, vilket gör att det nyaste inlägget alltid kommer högst upp Name är bara namnet på vem som skrev inlägget och message fungerar som området själva inlägget sparas i.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	<b>id</b>	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/> 2	<b>nimi</b>	varchar(255)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 3	<b>viesti</b>	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 4	<b>timestamp</b>	int(255)			No	None	

Bild 15. Gästbokens struktur i databasen.

Efter att jag lagt till den i databasen skapade jag själva gästboken. Den körs via PHP så den skickar information till databasen samt söker information därifrån till sidan.

Själva koden för funktionen som laddar upp är inte för komplicerad. Den består av två delar. Den visuella som innehåller själva uppladdnings layout och sedan koden som körs i bakgrunden av den (kod exempel 16).

*Kod exempel 15. PHP skapar rutorna som behövs för att fylla i information för databasen.*

```
<h3>Jätä kommentti:</h3>
<form action="post.php" method="post">
<strong>Nimi</strong><br/><input type="text" name="nimi" /><br/>
<strong>Viesti:</strong><br/> <textarea name="viesti" rows="5"
cols="55"></textarea><br/>
<input type="submit" value="Lähetä">
</form>
```

Koden ovanför som jag skapat är enbart layouten på själva funktionen där man lämnar inlägg i gästboken. Efter detta skapade jag resten som sköts i bakgrunden. Eftersom önskemålet är att få både namn och text till gästboken så gjorde jag det så att den alltid kontrollerar att det finns text i båda fälten när man laddar upp till databasen (kod exempel 17).

*Kod exempel 16. PHP skripten ser om alla fält är ifyllda.*

```
foreach ($fields as $field) {
if (empty($_POST[$field])) {
$isOkay = FALSE;
}
}
```

Efter att den har kontrollerat att alla fält är korrekta kör den sista delen av koden vilken kommer att skicka allt till databasen och placera det i de rätta fälten (kod exempel 18).

*Kod exempel 17. Skickar information till fälten nimi, viesti och timestamp*

```
if (mysql_query("insert into vieraskirja (`nimi`,`vesti`,`timestamp`
) values ('{$nimi}','$viesti','{$now}'))") {
```

När själva uppladdningskoden samt databasfälten är klara skapade jag delen som kommer att ta emot informationen samt visa den på hemsidan. Det som behövs är att den visar vem som skrev inlägget samt vad som skrevs och när det gjordes. För detta använder jag ECHO funktionen i PHP vilken hämtar information från de olika tabellerna i databasen och visar det på hemsidan (kod exempel 19).

Kod exempel 18. PHP skripten hämtar allt från databasen och visar det på gästboken.

```
require_once 'config.php';
$allPostsQuery = mysql_query("select * from vieraskirja order by
`timestamp` DESC ");
if(mysql_num_rows($allPostsQuery) < 1) {
ECHO "Vieraskirja on tyhjä!";
} else {
while($comment = mysql_fetch_assoc($allPostsQuery)) {
ECHO "<b>Nimi:</b> {$comment['nimi']} <br/>";
ECHO "<b>Viesti:</b> {$comment['viesti']} <br/>";
ECHO "<b>Lisätty:</b> " .date("Y-d-m
H:i:s", $comment['timestamp']). " <br/><hr/>";
}
```

I början av koden ser man `require_once` funktionen vilken kopplar upp till databasen före den kan göra något annat. Efter det tar den allt som finns inne i databasen under namnet `vieraskirja` och hämtar det till webbsidan. Ifall det är mindre än ett inlägg i databasen kommer den att säga ”Vieraskirja on tyhjä!”. Men ifall det finns något i databasen så visas det på hemsidan (se bild 16). I slutet har den kommandot `<hr>` som skapar ett sträck över sidan så att man lätt kan se var nästa inlägg börjar.

#### **Ismo**

Nam dictum, odio at laoreet cursus, lorem enim ullamcorper neque, a efficitur dui mauris ac tortor.  
2014-29-10 19:39:12

---

#### **Sami**

Pellentesque egestas ipsum in massa auctor aliquam.  
2014-29-10 19:38:57

---

Bild 16. Gästboks inlägg som de syns i webbläsaren.

## 6.6 Administratören

För att sidan skall vara så säker som möjlig och inte tillåta utomstående personer att editera, är den skapad med en administratör användare. Administratör användaren kommer åt vissa funktioner som inte är synliga för de vanliga besökarna. T.ex. kan administratören ladda upp nya bilder till galleriet, skriva nya inlägg på sidorna samt uppdatera de gamla sidorna.

Sidorna använder sig av samma template som vanligt men när man skriver in sig som administratör så länkas personen till sidor som har administrativa funktioner istället för till de vanliga sidorna.

När man besöker hemsidan skall adressen vara [www.kennelfinby.com/index.php](http://www.kennelfinby.com/index.php). Då man skriver in sig med de administrativa rättigheterna visar den sig som [www.kennelfinby.com/admin/index.php](http://www.kennelfinby.com/admin/index.php) eftersom administratör filerna finns i en annan mapp utanför de vanliga. Man kommer inte åt dem utan att logga in.

Försöker man komma åt administratör sidorna och inte är inloggad, kör den ett kommando som flyttar över till inloggningssidan (Se kod exempel 20).

*Kod exempel 19. PHP skripten kollar ifall man är loggad in som administrator.*

```
<?PHP
include_once 'admin-class.php';
$admin = new itg_admin();
$admin->_authenticate();
?>
```

Detta kommando kör admin-class.php filen som kontrollerar om man är inloggad eller inte på sidan genom att kolla med databasen. Om det kommer fram att man inte är inloggad men försöker komma åt administratör filerna, flyttar den över till login.php sidan som består av en ruta där man skall fylla i användarnamn samt lösenord.

Käyttäjänimi:

Salasana:

Muista minut

*Bild 17. Inloggnings ruta för administratören.*

Har man inte inloggnings rättigheter eller om man försöker kring gå, kör den koden som förflyttar över till inloggnings sida (kod exempel 21).

*Kod exempel 20. PHP skript som flyttar en till login.php sidan ifall man inte är in loggad.*

```
else {  
header("location: login.php");  
die();  
}
```

## 6.7 Logga in/ut

För att denna funktion skall vara så säker som möjlig, körs allt med PHP och med en MySQL databas där all information finns färdigt sparad. Det är inte möjligt att komma åt inloggningsinformationen om man inte känner till både användarnamn och lösenord till databasen. Eftersom hemsidan inte kommer att ha flera användare än administratören är den gjord så att det inte är möjligt att skapa flera administratörer utan att gå in i databasen direkt och där göra en ny.

Det enda som återstår att lösa är att skapa en knapp eller liknande för över till inloggningsrutor, samt möjlighet att logga ut vid behov. För tillfället måste man för att logga in skriva login.php och om man vill logga ut så måste man skriva in logout.php för att den skall logga ut användaren.

## 7 Testning av sidan

De flesta som skapar sidor testar sidan enbart för att hitta eventuella buggar så att de undviker extra arbete i ett senare skede. Detta är av stor betydelse men lika viktigt är det att testa sidan utifrån kundens behov och målsättningar. Uppfyller den inte kundens ursprungliga och även överenskomna krav, är det mycket troligt att de inte vill använda sig av erbjudna tjänster och program i fortsättningen. Ifall man levererar en produkt med många buggar och problem, betyder det att man i ett senare skede måste sätta ner extra tid på att lösa de problemen. Lyckas man genast bygga en produkt som fungerar och uppfyller kraven, ger det ett bra intryck av producenten samt kommer att kräva betydligt mindre tid för support i framtiden. Ju senare man hittar fel i produkten desto mera kommer det att kosta varför det är viktigt att man försöker finna dem redan i det skede sidan skapas.

(Eriksson, U. 2008, s.17-22)

Det är av stor betydelse att man genast i början av arbetet sätter tid på att identifiera risker och orsaker till eventuella problem, samt försöker beakta dem innan problemen uppstår. I detta arbete var en av det största riskerna att kunden skulle komma åt en del av koden av misstag och ändra på den och den vägen orsaka problem för själva sidan. Jag åtgärdade detta genom att göra områden som kunden kan komma åt på egen hand och låsa bort sådant som kräver hjälp av en någon som förstår sig på sidornas uppbyggnad samt funktioner. Ju mera jag begränsat de områden kunden själv kan komma åt, desto lättare har han hade att använda sidan utan att han riskerar ändra på layouten eller dylikt.

När det kom till testningen av sidan tog jag hjälp av familjemedlemmar och bad dem pröva använda sidan. Det första problemet uppstod vid inloggningen, då sidan inte hade en login knapp. Detta åtgärdades genast för att underlätta användandet. Som följande testades uppdateringen av inlägg samt uppladdning av bilder till galleriet. Detta fungerade utan problem. Även radering av inlägg fungerade utan problem. I ett senare skede kom det fram ett problem. Jag hade nollat ID värdet på varje tabell så att de måste börja från ett igen. Allt fungerade annars förutom att thumbnail filerna visade fel thumbnail bild. Den länkade nog ändå till rätt fil. Efter lite sökande kom jag fram till att problemet fanns i själva mappen där de sparades. Eftersom jag hade nollat ID talet men inte tagit bort de gamla thumbnail filerna, skapades det inte nya thumbnail filer, utan de gamla använde samma nummer som de nya borde ha använt. Efter att jag tömde mappen, uppstod det inga flera motsvarande problem.

## **8 Sammanfattning**

Nu när arbetet är klart är jag själv relativt nöjd med resultatet. Att bygga en hemsida från början till slut var ett mera krävande arbete än vad jag i början hade kunnat förutse. Det tog också längre tid än vad jag hade räknat med men under hela projektet har jag lärt mig en hel del genom alla problemlösningar mm som jag utsatts för. Jag är dock glad att jag började med detta projekt eftersom jag har lärt mig mycket nytt under arbetets gång samt återupplivat sådant jag glömt under åren som t.ex. databaser och hur de fungerar.

Det största problemet som uppstod och tog mest med tid, var att få själva thumbnail funktionen att fungera och skapa bilder. Jag försökte bygga den helt från grunden som en egen version, men då det tog så lång tid löste jag problemet genom att söka fram en färdig

kod som jag sedan använde mig av. På detta sätt fick jag problemet löst och arbetet att framskrida. En av de positiva sakerna med att arbeta med PHP och SQL var att det fanns så mycket information på internet att tillgå. Behövde jag en viss funktion eller om jag inte förstod exakt hur jag skulle göra, var det möjligt att studera sidor som w3schools och stackoverflow vilka har en enorm mängd av information vare det sig gäller PHP, SQL eller CSS eller annat. Där kunde jag hitta information från personer som råkat på motsvarande problem samt läsa hur de hade lyckats lösa problemen.

När jag presenterade den nya hemsidan och dess funktioner för kunden var han mycket nöjd med den slutliga versionen. Vi kunde tillsammans konstatera att de uppsatta målsättningarna uppfylldes och att den är enkel för kunden själv att uppdatera. Följande och kommande utveckling av hemsidan är att bygga en motsvarande sida där språket är svenska.

.

## Källförteckning

*The history of Photoshop. 2005. [Online]*

<http://www.creativebloq.com/adobe/history-photoshop-12052724> [hämtat:21.10.2014].

Faraon, M. & Sahlberg, M., 2012. *Webbutveckling med PHP och MYSQL.Lund:Studentlitteratur.*

msdn.microsoft *Structured Query Language (SQL)*(u.å.). [Online]

<http://msdn.microsoft.com/en-us/library/ms714670%28v=vs.85%29.aspx> [hämtat: 6.9.2014]

phpMyAdmin (u.å.). [Online]

[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php) [hämtat: 27.8.2014]

*jQuery-guiden* (u.å.).[Online]

<http://www.webbdesignguiden.se/jquery-guiden/>[hämtat:14.8.2014].

w3schools *Browser Display Statistics* (u.å.). [Online]

[http://www.w3schools.com/browsers/browsers\\_display.asp](http://www.w3schools.com/browsers/browsers_display.asp) [hämtat: 27.8.2014]

Walsh, D., 2008. *Create Image Thumbnails Using PHP.* [Online]

<http://davidwalsh.name/create-image-thumbnail-php> [hämtat: 3.9.2014].

Lindley, C.,2007. *Thickbox.* [Online]

<http://codylindley.com/thickbox/> [hämtat: 10.9.2014]

Eriksson, U., 2008. *Test och kvalitetssäkring av IT-system*

## Bildförteckning

Bild 1: Exempel på WinSCP användar gränssnitt.	2
Bild 2: WinSCP uppkopplings fönster.	4
Bild 3: Exempel på Dream Weaver Arbetsfönster	5
Bild 4: Händelseförloppet för PHP	6
Bild 5: Denna bild visar ett exempel på hur phpMyAdmin ser ut i webbläsaren.	9
Bild 6: Exempel på hur gamla sidan såg ut.	10
Bild 7: Gamla sidans menybalk	11
Bild 8: Information på skärmarnas upplösning samlad av w3schools	11
Bild 9: Området innanför den blåa rutan kan editeras, allt utanför är låst	12
Bild 10: Första design på den nya sidan	13
Bild 11: Layout på den nya hemsidan	16
Bild 12: KennelFinby databas struktur	17
Bild 13: Strukturen av index sidan i databasen	18
Bild 14: Exempel på galleriets struktur	23
Bild 15: Gästbokens struktur i databasen	25
Bild 16: Gästboks inlägg som de syns i webbläsaren.	27
Bild 17: Inloggnings ruta för administratören	29

## Kodförteckning

Kod exempel 1: PHP skript som visar ”Detta är en PHP-funktion!” när den körs	7
Kod exempel 2: Query skript som söker från pentuea tabellen och sorterar enligt namn	8
Kod exempel 3: CSS kod på område som kan editeras av användaren	12
Kod exempel 4: CSS hela hemsidan kommer att finnas innanför container område	14
Kod exempel 5: CSS de enskilda sidornas funktioner kommer att fungera innanför detta område	14
Kod exempel 6: Koden visar knapparna på actionbaren.	15
Kod exempel 7: PHP skripten för att skapafälten som används vid uppladdning av information	19
Kod exempel 8: PHP skript som sätter gräns på vilken filtyp som är tillåten	19
Kod exempel 9: PHP skripten ser ifall namnet är redan upptaget i databasen.	20
Kod exempel 10: PHP skripten söker original filen och kör make_thumb funktion som skapar thumbnail	20
Kod exempel 11: PHP Skript som skapar thumbnail (David Walsh kod)	21
Kod exempel 12: PHP skript som hämtar bildens information från databasen och visar den på index sidan	22
Kod exempel 13: Länkar thickbox.css till galleri sidan.	23
Kod exempel 14: PHP alla nödvändiga fälten för att koppla till databasen	24
Kod exempel 15: PHP skript som raderar inlägg baserat på ID nummer	24
Kod exempel 16: PHP skapar rutorna som behövs för att fylla i information för databasen	26
Kod exempel 17: PHP skripten ser om alla fält är ifyllda	26
Kod exempel 18: Skickar information till fälten nimi, viesti och timestamp	26
Kod exempel 19: PHP skripten hämtar allt från databasen och visar det på gästboken	27
Kod exempel 20: PHP skripten kollar ifall man är loggad in som administrator	28
Kod exempel 21: PHP skript som flyttar en till login.php sidan ifall man inte är in loggad	29