

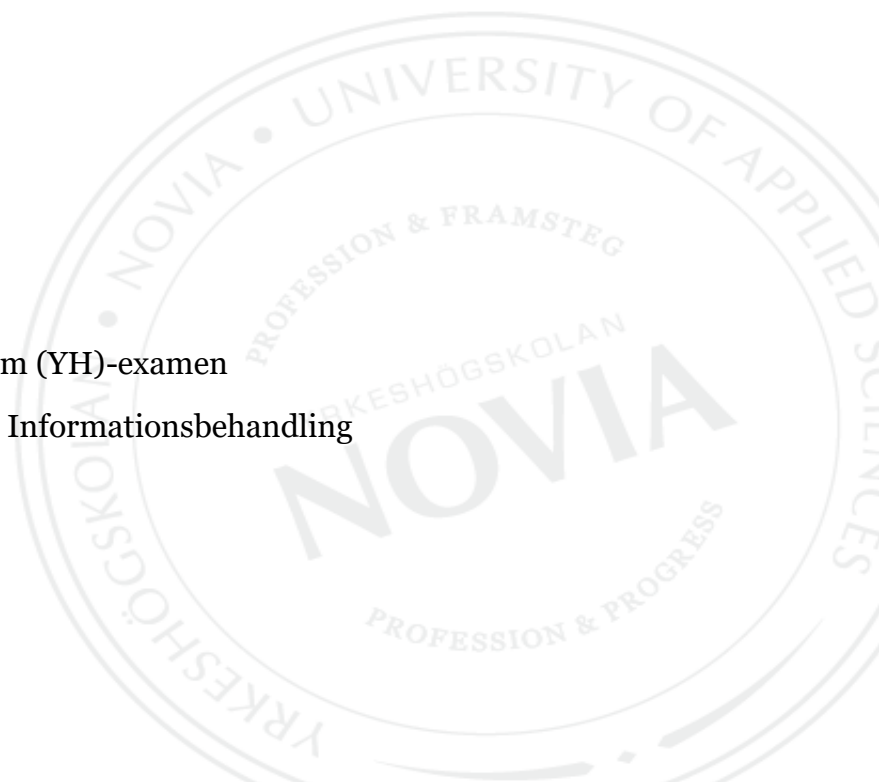
# **Utveckling av en enkel basmodell för en webbplats**

Wilhelm Javén

Examensarbete för Tradenom (YH)-examen

Utbildningsprogrammet för Informationsbehandling

Raseborg 2015



## EXAMENSARBETE

Författare: Wilhelm Javén

Utbildningsprogram och ort: Tradenom, Raseborg

Inriktningsalternativ/Fördjupning: Informationsbehandling

Handledare: Klaus Hansen

Titel: Utveckling av en enkel basmodell för en webbplats

---

20.11.2015

Sidantal: 31

Bilagor: 0

---

### Abstrakt

I detta examensarbete beskrivs hur man utvecklar en webbplats från grunden samt de olika steg som görs, från planering till implementering och slutförande. I arbetet behandlas även de vanligaste metoderna samt begreppen inom webbutveckling.

Webbplatsen byggs upp genom att först göra grunden för sidorna och sedan bygga in alla funktioner som används. Fokus i utvecklingen har varit att koda de olika funktionerna samt det tekniska arbetet med webbplatsen. Koden för funktionerna behandlas mera ingående eftersom det är viktigt för att förstå hur de är uppbyggda. Användarvänligheten är även en viktig del och därför har utseendet gjorts åskådligt.

Denna webbplats har gjorts för att fungera som en marknadsföringskanal och skall därför vara professionell och kunna fungera i alla skärmstorlekar och miljöer. Webbplatsen är lätt att hitta till samt att navigera på.

Denna webbplats är gjord så att man lätt skall kunna redigera och lägga till innehåll i framtiden. Det skall också vara möjligt att lätt kunna utöka webbplatsen och bygga in flera funktioner vid behov.

---

Språk: Svenska

Nyckelord: webbutveckling, marknadsföring, användarvänlighet,

kodning

---

## **BACHELOR'S THESIS**

Author: Wilhelm Javén

Degree Programme: Business Information Technology

Supervisors: Klaus Hansen

Title: Development of a simple model for a website

---

Date 24.11.2015

Number of pages: 31

Appendices: 0

---

### **Summary**

This thesis describes the process of making a website from start to finish. Explaining all necessary steps. The most used methods of making websites and the concepts in web development will be described.

When making a website the structure of the site is created first, then all the features and functions are added. In this project the focus is on the features and all the technical work. The coding of the features has been explained in more detail since it is important to understand how these are made. The usability is also an important aspect of the site.

This website will be used as a marketing site and therefore it has to be professionally made and usable regardless of screen size or environment.

The website is made so that the content easily can be updated or changed. It will also be possible to expand the site by adding more features if necessary.

---

Language: Swedish

Key words: Web development, marketing, usability, coding

---

## Innehåll

1. Inledning.....	1
1.2. Syfte.....	1
1.3. Bakgrund .....	2
2. Metoder.....	2
2.1. Val av metoder.....	3
2.1.1. Joomla.....	4
3. Marknadsföring .....	8
3.1. Sökmotoroptimering.....	8
4. Uppbyggnad .....	10
4.1. Allmänt .....	10
4.2. Begrepp.....	10
4.3. Förberedelser .....	12
4.3.1. Verktyg .....	12
4.4. Grafisk Design.....	13
4.5. Tekniker.....	14
4.5.1. CSS.....	14
4.5.2. PHP.....	16
4.5.3. HTML.....	16
4.5.4. Javascript .....	16
4.5.5. SQL.....	16
4.5.6. XAMPP .....	17
4.6. Webbhotell och domän.....	18
4.7. Filstruktur .....	18

4.8. Hämtning av databaser .....	19
5. Webbsidans design .....	20
5.1. Val av layout.....	20
5.2 Responsivitet .....	21
6. Förverkligande av webbplatsen .....	22
6.1. Kodning .....	23
6.2. Inloggning.....	24
6.3. Säkerhet .....	25
6.4. Formulär .....	26
6.4.1. CAPTCHA .....	27
6.5. Språk.....	30
7. Slutsats.....	30
8. Sammanfattning.....	31
Källförteckning.....	32

## 1. Inledning

Webbutveckling är idag en bred bransch där det finns många olika tekniker och verktyg att använda. Det är en bransch som ständigt utvecklas och behovet av webbutvecklare blir allt större. Webbutveckling omfattar allt som har med internet och webbsidor att göra. Webbsidor kan idag se väldigt olika ut och måste fungera i många olika typer av enheter och skärmar.

Detta arbete beskriver hur man skapar en enkel fungerande webbplats från grunden teoretiskt. Jag kommer också att skapa en webbplats åt mig själv. Jag kommer att gå igenom alla olika steg så att man bättre förstår hur webbplatsen är uppbyggd.

Varför jag har valt just detta arbete beror främst på att jag är intresserad av att koda webbsidor men också att själv kunna marknadsföra mig på internet genom en professionell webbplats som alla kan komma åt och titta på.

Från början försökte jag hitta någon som behövde en webbplats som jag skulle kunna utveckla en åt, men då jag inte lyckades hitta någon så bestämde jag mig för att utveckla en webbsida åt mig själv istället. Detta gör att det blir en utmaning då man själv måste fundera ut hur webbplatsen skall se ut och vad som skall behövas finnas på sidan. Det blir också intressant att se resultatet och framför allt att se nyttan man har av den.

### 1.2. Syfte

Syftet med detta arbete är att visa hur man skapar ett enkelt skal för en webbplats. Detta blir en god erfarenhet för mig. Jag räknar med att jag bättre kommer att förstå kod samt nyttan med koden och dess funktion. Jag kommer även att få en god insikt i hur webbplatser överlag är uppbyggda. Jag får alltså en bättre överblick över webbplatsen då jag själv bygger upp den och har möjlighet att senare lätt lägga till och redigera kod i efterhand.

Webbskalet omfattar ett administrativt gränssnitt för uppdatering av innehåll, formulär med Captcha, ett användarvänligt gränssnitt, responsivitet samt olika språkversioner av webbplatsen.

### 1.3. Bakgrund

Idén till detta arbete började egentligen för flera år sedan. Jag har redan länge varit intresserad av att utveckla en professionell webbplats. Därför känns det bra att jag kan ägna detta arbete åt att skapa en webbplats som jag gör åt mig själv och som jag har nytta av och kommer att kunna använda i framtiden och dessutom lära mig att hjälpa andra. Jag har tidigare skapat enkla webbsidor åt andra och hjälpt till att publicera saker på webben.

Jag har länge funderat på hur en bra webbsida skall vara och hur den skall se ut. Jag har studerat många andra webbsidor och vi har också haft många webbprojekt i skolan, mot denna bakgrund skall jag göra detta arbete.

## 2. Metoder

När man utvecklar webbplatser finns det några olika metoder och verktyg som man kan använda. Om man vill skapa en webbplats men inte vet så mycket om webbutveckling och hur man skall göra skall man börja med att leta efter information på internet. Man kommer då ganska snabbt att hitta bra guider för webbutveckling och olika verktyg.

På internet finns mycket kod som man kan använda och sätta upp på ett webbhotell för att få en fungerande webbplats. Man kan också kombinera olika metoder för bästa resultat.

Man kan även använda färdiga stilmallar om man har svårt att bestämma sig för hur webbplatsen skall se ut.

Adobe Flash användes mycket förut för att göra publikationer eller olika animationer med på webben men då flash inte mera används lika mycket nu på grund av att HTML5 och css3 har blivit så väl utvecklade så känns flash inte mera som ett bra alternativ att använda. Flash har dock den fördelen att det är ganska lätt att lära sig och att använda.

Ett av de vanligaste sätten att skapa webbsidor är att använda sig av ett CMS (Content Management System). CMS är ett webbpubliceringsverktyg som gör det enkelt att skapa innehållet på en webbplats. CMS körs direkt från en webbserver. Det finns många olika CMS system men de tre vanligaste är Joomla, Drupal och Wordpress.

## 2.1. Val av metoder

När jag hade bestämt mig för att utveckla en egen webbsida började jag genast fundera på vilken metod jag skulle använda mig av. Även om jag har ganska bra koll på vad det finns och på hur man skapar en webbsida så måste jag ändå avgöra vad jag skulle välja och hur jag skulle arbeta.

Eftersom jag har tidigare erfarenheter av CMS: et Joomla så funderade jag först om jag skulle kunna använda det och bygga min webbplats i Joomla. Jag anser personligen Joomla är bra då det är ett webbpubliceringsverktyg som är mycket åskådligt uppbyggt och därför underlättar skapandet av innehåll. Det kräver inte lika mycket som t.ex. att skapa webbsidor i Drupal. Joomla är också lätt att lära sig. Jag tog också olika ramverk i beaktande om det vore möjligt att använda något av dem. Fördelen med ramverk är att de innehåller färdiga funktioner för att skapa olika applikationer eller dynamiska webbsidor, men jag valde bort den möjligheten. (Jeffknupp, u.å.).

Efter att ha funderat på olika möjligheter så bestämde jag mig för att bygga upp hela webbplatsen själv genom kod då jag vill få erfarenhet av kodning för att bättre kunna hantera och förstå webbutveckling. Det är mycket mera utmanande att använda och modifiera kod själv. Jag har tidigare använt andra metoder och bara gjort enkla script så det är därför jag nu vill skapa en hel webbplats själv från grunden.

En orsak till valet är också att jag är verkligt intresserad av det tekniska inom webbutveckling och därför väljer att skapa webbplatsen själv, istället för att bygga upp något i grafisk miljö eller via något annat verktyg.



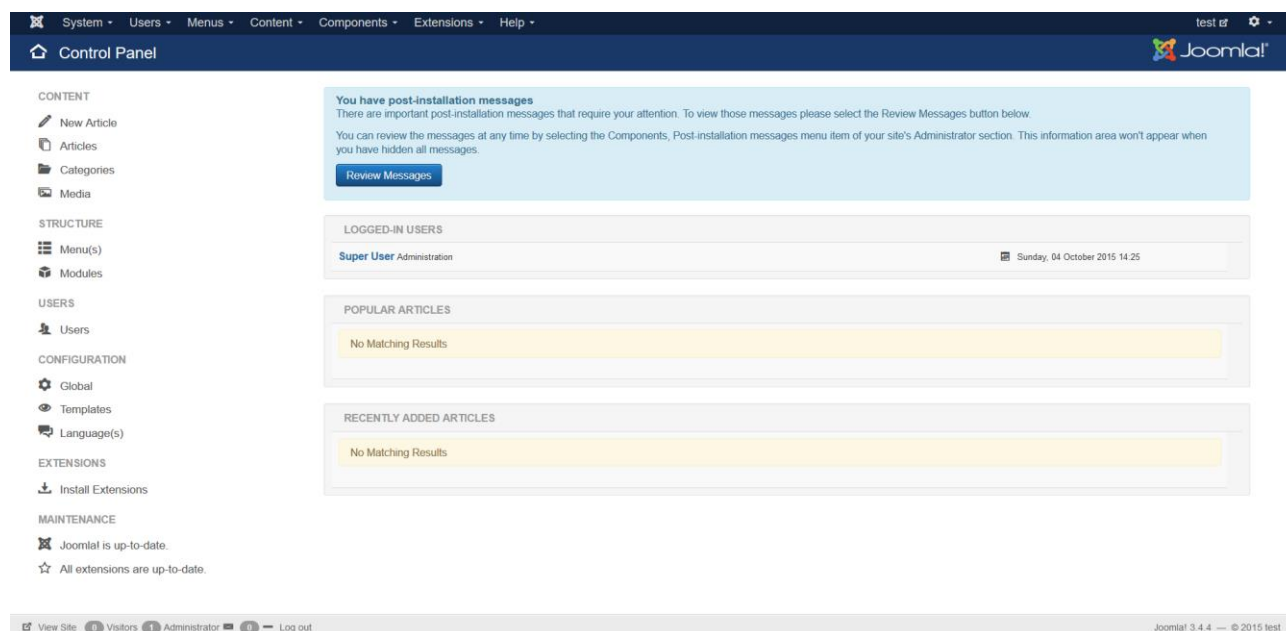
### 2.1.1. Joomla

Jag kommer nu att närmare behandla Joomla och dess funktionalitet, eftersom jag tidigare har gjort webbsidor med Joomla.

Om jag hade gjort den webbplats som jag skapat i Joomla eller andra verktyg för webben så hade jag förmodligen använt en liknande layout och skulle säkert ha byggt upp ett liknande koncept motsvarande det jag nu gör.

För att kunna använda Joomla måste man först installera det. Man måste ha en server att sätta upp Joomla på innan man installerar. Man måste också först skapa en databas då Joomla lagrar information i databaser. Under installationsfasen anger man namn på sidan och skapar ett konto att logga in med. Man väljer också olika inställningar för Joomla. Installationen är enkel och går snabbt.

Joomla består av ett frontend och ett backend gränssnitt. Backend används för själva skapandet av webbplatsen medan frontend är den sidan man ser då man besöker sidan. I backend miljön finns alla verktyg och inställningar för själva sidan (Bild 1). I frontend kan man lägga till allt man vill att skall synas på sidan. Man kan också enkelt redigera innehållet i frontend.

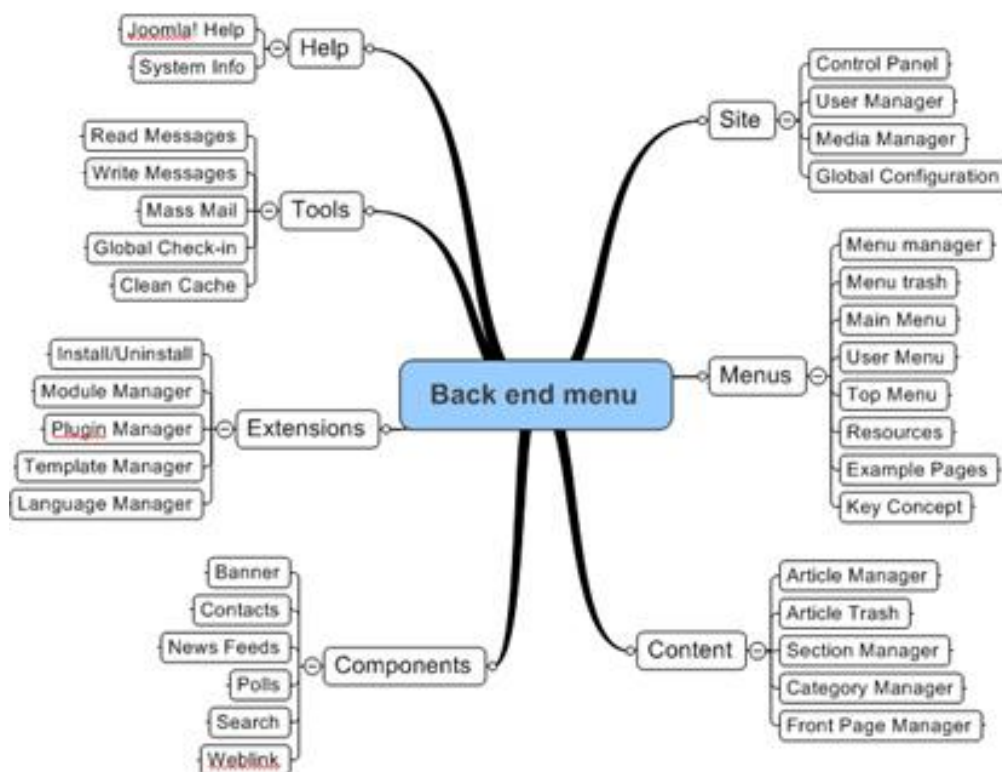


**Bild 1. Bild över backend i Joomla**

För att komma åt backend skriver man in administrator efter domännamnet i adressfältet i webbläsaren. Då kommer en inloggningsruta där man loggar in och kan efter det börja jobba i Joomla. Då man börjar bygga webbplatser med Joomla så börjar man med att först skapa olika kategorier för innehållet samt de olika moduler som består av de olika funktionerna på webbplatsen som sedan kan placeras ut där man vill ha dem på sidan.

Joomla innehåller från början stöd för hantering av flera språk samt hantering för olika media. Det är även möjligt att enkelt behandla de stilmallar man använder och övriga inställningar för webbsidan. Man kan t.ex. välja position på de olika objekten på sidan samt var de skall visas. Det finns även mängder av olika tillägg och stilmallar som gör att man får ytterligare funktionalitet och kan skapa mera avancerade webbsidor.

För att lättare förstå hur backend i Joomla är uppbyggt skall man studera en bild över strukturen (Bild 2). (Pheveryday, u.å.).



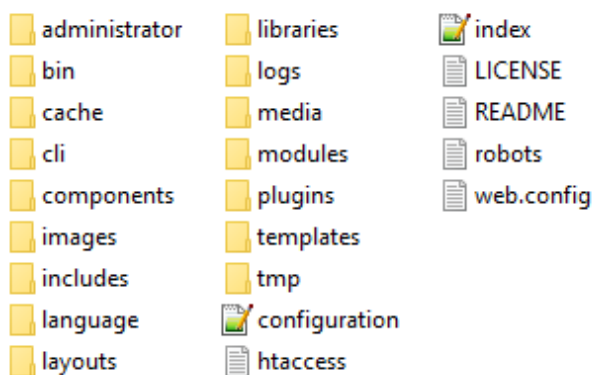
**Bild 2. Struktur över Backend i Joomla**

Det som gör Joomla så bra och lätt att använda till skillnad från de övriga CMS systemen, är att Joomla har ett mycket bra användargränssnitt där man enkelt hittar alla inställningar och lätt kan skapa innehåll. Joomla är lätt att komma igång med och det går lätt att skapa en fungerande webbplats.

Joomla har också ett bra verktyg att hantera användare med. Man kan enkelt skapa flera användare och tilldela skilda rättigheter åt dem alla.

Det går också lätt att logga in direkt via frontend och därifrån redigera innehållet utan att behöva använda backend. Detta är en fördel med Joomla att inloggningen är enkel och säker från början.

För att kunna använda Joomla mera avancerat och skapa egna tillägg eller stilmallar måste man förstå hur Joomla är uppbyggt. Joomla består huvudsakligen av PHP kod som styr de olika funktionerna i Joomla. Det finns skilda mappar för de olika funktionerna (Bild 3). Frontend i Joomla är stilmallar som man lägger in i en separat mapp i Joomla.



**Bild 3. Joomla's filstruktur**

I administrator mappen finns alla filer som behövs för backend. Det är gränssnittet där alla inställningar finns samt därifrån allting i Joomla styrs. I plugins mappen finns alla tillägg, I template mappen finns alla stilmallar. Om man skapar en egen stilmall läggs alltså denna hit. De övriga mapparna används av Joomla's övriga funktioner. Man kan lägga till filer manuellt i Joomla, men då måste man veta var de skall placeras samt vad det man sätter till har för funktion. (Brasil P, 2012).

För att få en webbplats i Joomla responsiv som fungerar i olika miljöer så använder man oftast färdiga stilmallar som fungerar från början. Det finns även separata stilmallar för mobilskärmar som man kan använda. Det är alltså mycket enkelt i Joomla att skapa fungerande webbsidor. Det finns många olika typer att välja mellan, men om man ändå vill skapa en egen stilmall från grunden så behöver du ha Joomla installerat och köra de filer du skapar via Joomla eftersom stilmallen används i samband med Joomla.

För att skapa en grundstilmall till Joomla behövs egentligen bara tre filer, en index.php fil som innehåller kod till Joomlas alla funktioner. Den andra filen är en XML fil som innehåller information om mallen som Joomla behöver veta såsom namn, beskrivning av mallen och olika positioner som skall finnas på sidan. Man behöver givetvis också ha en css fil för layout och utseende. (Rohde M, u.å.).

Joomla används idag mycket av professionella webbutvecklare då det är mycket flexibelt och är fritt att använda. (Joomla, u.å.).

För den som inte har så mycket erfarenhet av webbutveckling överlag så lönar det sig att först lära sig koda ordentligt för det är viktigt att man kan skapa lite mera avancerade script förrän man börjar använda andra verktyg. Det är lättare att komma igång med Joomla eller övriga webbpubliceringsverktyg om man först förstår koden så att man kan bygga upp webbsidor från grunden. Man kan också hantera webbutvecklingsverktyg bättre då du kan koda eftersom all webbutveckling är i grunden baserad på kod.

Nackdelen med att använda de metoder som har behandlats i detta arbete eller andra verktyg som finns är att det kan vara svårt att komma igång och att lära sig dem. Många verktyg kräver också att man har kunskaper från tidigare. Det finns dock mycket information på internet om hur man skall använda dessa verktyg.

Man har bättre överblick över webbplatsen om man själv skapar kod och bygger upp än om man skulle använda andra verktyg eller färdiga stilmallar. Det går även mycket lätt att redigera och lägga till kod då man vet vad man har gjort.

### 3. Marknadsföring

Marknadsföring är en viktig del av all webbutveckling. Det är viktigt att se till att man själv eller att ens företag syns på internet så att andra enkelt kan hitta till ens företag. När jag utvecklar min egen webbplats gör jag det i syfte att marknadsföra mig själv som yrkesmänniska.

Konkurrensen är idag hård inom IT-branschen och därför är det viktigt att man ser till att man har ett gott rykte och att man finns på internet. Man bör också kunna visa sin kompetens. Det kan man göra bl.a. genom en egen webbplats där man lägger upp information och material som andra kan hitta. Det är också därför som jag själv vill koda min webbplats, för att genom den visa att jag behärskar källkoden och förstår hur webbplatser är uppbyggda.

Även om man i framtiden inte skulle jobba direkt med webbutveckling så är det alltid bra att ha en egen webbplats där man kan samla material och där andra kan ge feedback.

Det viktigaste med en personlig webbsida är att den är så bra och informativ att man själv kan vara stolt över den. Det viktigaste är att den ger en positiv och seriös bild av dig som IT - person.

#### 3.1. Sökmotoroptimering

Det är viktigt att det går lätt att hitta till min webbplats. Man strävar efter att hitta sidan högst på Google om någon söker på ens namn eller söker på saker som man har gjort. Detta kallas för sökmotoroptimering och betyder att man optimerar sin webbplats med unika titlar eller nyckelord som används på webbplatsen. För att göra detta använder man olika meta tags i Html. I meta taggar anger man information om webbsidan och gör olika beskrivningar som sedan hittas av Google (Kod 1).

##### ***Kod 1. Exempel på meta tags***

```
<meta charset="UTF-8">  
<meta name="description" content="beskrivning av webbsidan">  
<meta name="keywords" content="innehållet I webbsidan">
```

I description taggen beskriver man kort webbsidans innehåll med några ord så att en sökmotor enkelt kan hitta sidan med dessa ord. Det är viktigt att man beskriver webbsidan korrekt med relevant information över det som finns på den sidan. Med keywords taggen ger man nyckelord för sidan som man vill att skall hittas av sökmotorn när sökordet innehåller något av dessa.

Man kan använda <title> taggen för att skapa en unik titel för en webbplats som då kommer att hittas av sökmotorn när man söker på den titeln.

Man bör ha beskrivande kategorier och filnamn på dokument som man har på webbservern, eftersom detta underlättar för sökmotorerna då de går igenom dina dokument. Dessutom blir webbadresserna mer användarvänliga om man har korta och beskrivande filnamn som också är lättare att länka till. Långa webbadresser med mycket tecken är jobbiga att avläsa och att använda. Webbadressen visas i Google nedanför titeln på sidan och då är en kort och relevant webbadress mera intressant för användaren.

Innehållet på en webbplats måste vara tilltalande för att få besökare att vilja stanna på sidan. Därför är startsidan viktig då det är den som visas först när man kommer till webbplatsen. Man bör regelbundet uppdatera innehållet på sidan eftersom sökmotorerna upptäcker när ny information kommer som sedan hittas i sökresultaten. Om man har gammal information som inte är uppdaterad kan det hända att sökmotorerna inte hittar den informationen lika bra som ny information.

Ett annat sätt att hålla sig synlig är att länka till en webbplats från andra sidor. Detta ökar synligheten för webbplatsen.

Det är också viktigt att sidan är lätt att navigera på för då hjälper det besökare att snabbt hitta det innehåll de söker. Det är också bra om sökmotorn vet vilken sida som är viktig på webbplatsen. Därför kan man göra upp en sidstruktur över webbplatsen så Google vet vilka sidor som prioriteras. (Bebetteronline, u.å.)

## 4. Uppbyggnad

### 4.1. Allmänt

Webbplatsen är kodad med php och html samt css för layout. Sidan använder sig också av Mysql databaser för lagring av text. En webbserver gör att sidan är lätt komma åt från olika platser. Då är man inte heller bunden till en och samma dator. Sidan skall ha responsiv design vilket betyder att den fungerar i alla skärmstorlekar.

### 4.2. Begrepp

Här kommer en lista på olika begrepp som används inom webbutveckling.

#### **Webbhotell**

För att kunna ha en webbsida på Internet måste man ha ett webbutrymme på en server. Det finns många olika webbhotell som erbjuder olika möjligheter för ens behov. På webbhotellen kan man också registrera ett domännamn.

#### **Domän**

Alla webbsidor och e-poster har ett domännamn t.ex. [www.google.com](http://www.google.com). Detta för att lättare kunna komma åt sidor istället för att behöva skriva in en IP adress varje gång som t.ex. 64.233.161.147. Varje domännamn är kopplat till en egen IP-adress.

Varje domän kan också ha flera underdomäner och varje underdomän kan i sin tur ha flera underdomäner osv. Underdomäner används främst av företag som har olika avdelningar och kan därför behöva delas in i olika namn. Man bygger alltså upp en hierarki på en server.

#### **FTP**

File Transfer Protocol är en teknik som används för att enkelt flytta filer mellan en dator och en server. När man har en server kan man specificera vad användaren skall kunna komma åt via ftp. Oftast har varje användare sitt eget område dit man laddar upp filer. I en webbserver kommer man oftast bara åt sitt eget webbutrymme och detta är viktigt på grund av säkerheten.

Kommunikationen med ftp anses vara ganska osäker eftersom det inte krypterar informationen mellan klient och server. Därför använder man oftast SSH (Secure Shell) i samband med ftp för att göra överföringen säker.

Man måste ha ett FTP klientprogram för att kunna komma åt filerna på en server. Det finns många olika ftp program t.ex. Filezilla, Winscp och ett flertal andra.

## **Databas**

Databaser används för att samla och lagra information på en server eller webbplats där informationen är lätt att organisera och hantera. I databaserna kan man ha olika tabeller med skild information så att man vet vart de hör. Informationen i databasen kan sedan hämtas in och läsas till webbplatsen. Databaser är ett av de bästa sätten att lagra information på och används därför mycket på webbplatser.

## **Kryptering**

Kryptering betyder att informationen omvandlas till tecken som inte säger den som läser något om innehållet. Man har genom historien använt ett antal olika metoder för att inte obehöriga skall komma åt viktig information.

Inom datavärlden används olika former av algoritmer för att skapa andra tecken. Programmet eller servern kan dock förstå den informationen som vi människor inte förstår.

## **SSL**

Secure socket Layer är en säkerhetsmetod som används för att kryptera kommunikation mellan en server och en klient. Vanligtvis används SSL i webbutiker eller på bank sidor. För att använda SSL behöver man ha ett certifikat. (Digicert, 2015).

SSL är vanligtvis aktiverat från början på de flesta webbhotell och då räcker det med att skriva https:// i början av adressfältet för att göra sidan säker.



## 4.3. Förberedelser

Det första man gör när man börjar planera en webbplats är att fundera ut vilka funktioner som skall finnas med. Det är bra om man före arbetet också har en klar idé om hur webbplatsen kommer att se ut. Det lönar sig att ta sig tid att skriva ner idéer och börja planera innehållet i arbetet innan man sätter igång.

Man skall också fundera ut syftet med en webbplats och vilken målgrupp den har. Det finns olika projektmetoder som man jobbar med när man arbetar med projekt. Det är lättare under projektets gång att arbeta om du har en väl utförd plan att gå efter. När man börjar med själva arbetet är det lättare om man på förhand har sökt upp vilka verktyg som skall användas. Man kan börja med att först testa att skapa några enkla script för att se hur det fungerar.

Arbetet underlättas om man färdigt söker upp kod från internet på det man tänkt göra så att man vet ungefär hur det kommer att bli. Det är viktigt att man vet vad som krävs när man börjar utveckla en webbplats, annars kan det hända att man inte vet hur man skall göra eller att den blir halvfärdig.

De funktioner som jag åtminstone till en början planerar att bygga in på min webbplats är hämtning av databaser, kontaktformulär, information om mig själv, cv och inloggning. Jag kommer i framtiden att uppdatera sidan vart efter jag hittar nya saker som jag vill ha med på sidan.

För att få upp webbplatsen och få den att synas på internet måste man ha ett domännamn och en server. Jag hade från början tillgång till ett webbutrymme som jag har skaffat från tidigare så det var bara att sätta igång med utvecklingen.

### 4.3.1. Verktyg

För detta arbete används följande verktyg:

Winscp är ett ftp program som man använder för att flytta filer mellan datorn och servern. Man kan också redigera koden direkt från servern. WinScp har också en inbyggd terminal om man hellre vill använda kommandon för servern.

PhpMyAdmin är ett grafiskt användargränssnitt för hantering av mysql databaser som gör det enkelt att lägga till databaser eller ändra i befintliga databaser. PhpMyAdmin körs direkt i webbläsaren. Man måste logga in på sitt webbhotell för att komma åt PhpMyAdmin.

Notepad++ är en liten och enkel textredigerare som man skriver kod med och som stöder de flesta programspråken.

Firebug är ett litet men kraftfullt tillägg till Firefox som gör det möjligt att enkelt hitta och ändra koden för en webbsida direkt i webbläsaren utan att det sparar ändringarna i filerna. Om du t.ex. bara snabbt vill kolla ett element på en webbsida och vill kunna ändra i koden för att se hur det ser ut i webbläsaren skall man använda Firebug.

#### 4.4. Grafisk Design

När det kommer till webbdesign finns det egentligen tre olika typer av design. Jag kommer att gå igenom dem kortfattat och redogöra skillnaden mellan dem.

Till grafisk design tillhör allt det grafiska utseendet för en webbsida. När man designar en webbsida bör man fundera ut vilka färger och typsnitt man vill använda. Man bör också tänka på vilken målgrupp webbsidan kommer att ha och fundera hur sidan skall se ut för just den målgruppen. Det är inte svårt att göra en webbsida eller att hitta ett bra utseende. Däremot är det svårare att få en webbsida att verkligen fungera smidigt. Man måste tänka på att webbsidan skall fungera i många olika miljöer och att den ser då likadan ut. En webbsida bör vara lätt att använda och lätt att komma åt och det skall vara lätt att hitta det man söker. Den får inte heller vara tungladdad, d.v.s. för många bilder eller annat som gör att webbsidan blir trög. Därför måste bilder vara komprimerade på rätt sätt och färgerna rätt valda så att de passar ihop med bilderna.

Målet med grafisk design är att de som besöker sidan gillar det de ser så att informationen går hem.

## **Informationsdesign**

En webbsida behöver också en informationsdesign. Till skillnad från grafisk design så handlar informationsdesign om att hitta och begripa det som finns på en webbsida. Det är viktigt att organisera det som finns på en webbsida samt att fördela innehållet på sidan rätt.

Det handlar också om att kunna formulera informationen rätt så att besökaren enkelt förstår den information som finns på sidan.

## **Interaktionsdesign**

Den tredje typen av design är interaktionsdesign som kort och gott handlar om att göra och handla. Formulär på en webbsida t.ex. är interaktion. Besökaren på sidan fyller i information eller skickar meddelanden. Med interaktionsdesign måste man också tänka på i vilken ordning interaktioner ska ske. Feedback och respons är alltså en viktig del. Kan det resultera i att det blir felaktiga meddelanden av besökaren?

De flesta webbsidor innehåller alla tre typer av design. Det gäller att hitta en balans mellan de olika design stilarna. Det förekommer sidor som nästan bara består av grafisk design som t.ex. reklamsidor eller flash-sidor. (Söderström J, 2001).

## **4.5. Tekniker**

Följande kapitel behandlar allmänt de olika tekniker som används då man skapar en webbsida. Det finns en mängd olika tekniker som man använder sig av inom webbutveckling och dessa skall behandlas här.

### **4.5.1. CSS**

Css (Cascading Style Sheet) används för att forma webbsidor med hjälp av olika attribut. Med Css kan man enkelt ändra en webbsidas utseende genom att lägga till eller ta bort en textsträng. Fördelen med Css är att samma mall kan användas för flera sidor. Detta gör det enkelt att i efterhand ändra utseendet om man vill eller behöver. Css läses in direkt i webbläsaren och därför kan olika webbsidor variera lite i utseendet eftersom varje webbläsare tolkar css lite olika.

Css används ofta som externa filer som hämtas in med HTML. Detta för att göra koden mera överskådlig och för att minska på koden i html dokumentet. Det går även att skriva Css direkt i ett html dokument.

Css används i samband med HTML och PHP. För att html skall hitta css används en tag (Kod 2).

***Kod 2. Länk till css I html.***

```
<link rel="stylesheet" type="text/css" href="Css/test.css" />
```

Den nyaste versionen av css är css3 som man använder för att animera eller lägga till effekter på en webbplats. På min webbplats har jag css3 för att få menyerna att byta färg långsamt när man för musen över dem samt att texten kommer in långsamt när sidan laddas in.

När man animerar med css 3 använder man sig av så kallade keyframes där man anger vad som skall hända. (Kod 3).

***Kod 3.exempel på css3 animation.***

```
@keyframes fadein {
  from {
    opacity: 0;
  }
  to {
    opacity: 1;
  }
}
```

I detta kodexempel så har vi keyframes fadein där vi sedan anger vad som skall hända d.v.s. i det fall gå från opacity 0 till opacity 1 vilket betyder att den går från att vara osynlig till synlig. Efter att ha skapat själva funktionen måste man lägga in det i det element som man vill att det skall ske på: animation: fadein 2s; Här vet css att den tar animationen fadein och att animeringen tar 2 sekunder.

Css3 används också för att skapa gradienter på färger och skuggeffekter på texter eller andra objekt. Det finns även mycket mera man kan skapa med css3. Med css3 kan man lätt skapa dynamiska webbsidor. Css3 utvecklas hela tiden och flera funktioner tillkommer.

Syftet med css3 är att man lättare skall kunna skapa visuella effekter och bättre användargränssnitt med mindre kod. Tidigare gjordes animationer eller effekter med javascript eller med flash och var betydligt mera krävande att göra. Css3 laddas in snabbare av webbläsaren och tar inte så mycket resurser av datorn som med tidigare metoder. (Goodwin R, 2013).

### 4.5.2. PHP

PHP står för hypertext preprocessor och är ett populärt scriptspråk som används för att skapa dynamiska och interaktiva webbsidor såsom olika formulär, inloggning eller hantering av databaser. PHP exekveras på webbservern, vilket betyder att det är ett serverbaserat språk som bara kan användas i servermiljö.

### 4.5.3. HTML

HTML(Hypertext Markup Language) är det språk som man bygger upp strukturen på en webbsida med. Html använder sig av så kallade taggar, där varje element på sidan har en skild tag. HTML används tillsammans med javascript, PHP och Css.

### 4.5.4. Javascript

Javascript är ett scriptspråk som är mycket vanligt för att skapa olika typer av interaktiva webbsidor. Till skillnad från PHP är Javascript inte serverbaserat vilket innebär att det körs direkt i webbläsaren. Javascript skrivs direkt i HTML koden så man behöver inte ha några skilda filer eller program för att använda Javascript. Man kan också skriva javascript i separata filer som hämtas in av html.

### 4.5.5. SQL

SQL (Structured Query Language) är ett frågespråk som man använder för att ställa frågor till en databashanterare (McCarthy-Parton, T, 2005). SQL används av alla databaser idag men det finns olika typer av databashanterare som skiljer sig lite från varandra.

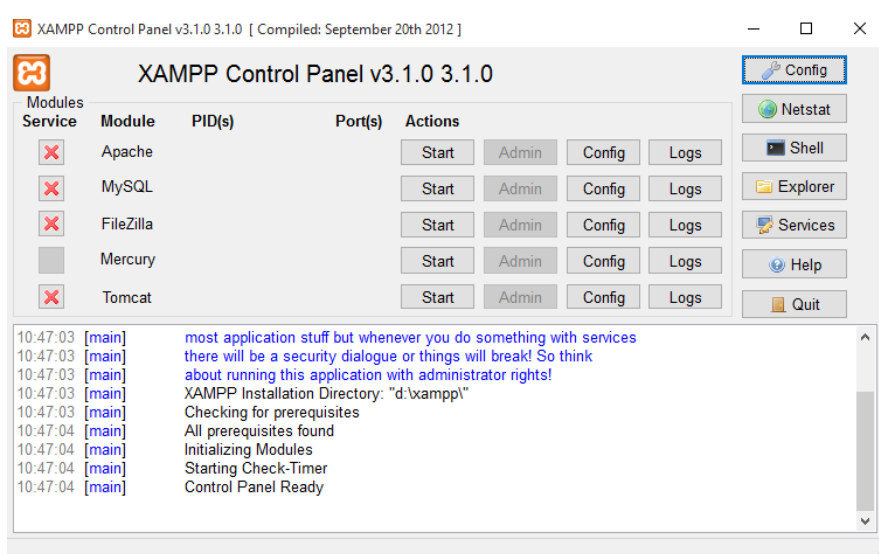
Den kanske mest populära databashanteraren är MySQL som är gratis och relativt enkel att använda. MySQL kan t.ex. användas tillsammans med PHP för att hämta information från en databas till webbsidan. En annan databashanterare som används mycket numera och som är gratis att använda är PostgreSQL. Denna är till skillnad från Mysql betydligt mer avancerad och har alltså flera funktioner än MySQL. (Susnet, u.å.).

#### 4.5.6. XAMPP

XAMPP är ett program med öppen källkod som fritt kan användas och som gör det möjligt att köra en webbserver på ens egen lokala dator utan att man behöver vara uppkopplad till internet. Syftet med detta är att man enkelt skall kunna testa webbsidor eller andra webbapplikationer direkt från sin dator innan man lägger upp dem på en riktig webbserver.

XAMPP består av Apache som är en webbserver, MySQL som är en databas hanterare, PHP som är ett scriptspråk och perl som också är ett scriptspråk. Xampp innehåller också PhpMyAdmin som gör att man enkelt kan hantera databaser. Detta betyder att du får en fullt funktionell webbserver med stöd för de flesta språk och verktyg som finns på en webbserver. Xampp är också mycket enkel att installera och använda.

För att starta webbservern så startar man Xampp kontroll panelen. I kontrollpanelen hittar du de tjänster som finns i Xampp. Härifrån kommer man också åt konfigurationsfilerna för tjänsterna. (Bild 4).



**Bild 4. Xampp kontrollpanel**

XAMPP körs från en webbläsare genom att skriva in i adressfältet 127.0.0.1 eller localhost vilket syftar på att det är en lokal server. (Apachefriends, 2015).

## 4.6. Webbhotell och domän

Webbhotellet jag använder är one.com som är ett billigt webbhotell där man får tillgång till det mesta som man behöver för att lägga upp och ha en fungerande webbsida.

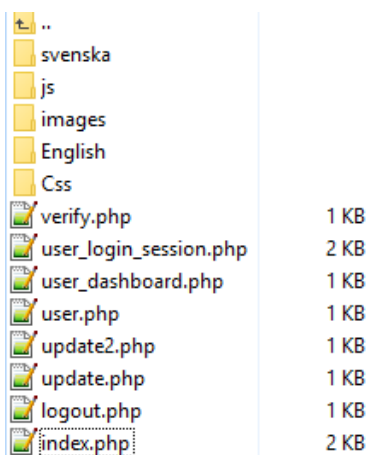
Domännamnet som jag valt för min sida är wilhelmjaven.com och det har jag ju valt eftersom det är min personliga webbsida så att man lätt kan associera till mig och mitt namn och för att lätt hitta mig på internet.

Visserligen kan man ju tycka att det skulle vara bra att använda sig av .fi istället men de flesta nationsdomäner kostar extra att registrera.

## 4.7. Filstruktur

För att ha en bra ordning på servern och för att lätt hitta alla filer har jag gjort upp en filstruktur. Jag har skilda mappar för bilder, css och javascript samt mappar för språken.

index.php är startsidan, den sida som visas när man besöker webbsidan. De övriga php filerna innehåller de funktioner som finns på webbplatsen. (Bild 5)



..	
svenska	
js	
images	
English	
Css	
verify.php	1 KB
user_login_session.php	2 KB
user_dashboard.php	1 KB
user.php	1 KB
update2.php	1 KB
update.php	1 KB
logout.php	1 KB
index.php	2 KB

**Bild 5. Filstrukturen på servern**

## 4.8. Hämtning av databaser

På min webbplats använder jag mig av databaser för lagring av text. Jag har en databas att använda mig av från webbservern men man kan lägga till flera tabeller som kan användas skilt för olika sidor. Jag har en tabell för varje text som jag har på sidorna. I varje tabell kan man också lägga till så många fält som man vill ha (Bild 6).

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> index	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> info	Browse Structure Search Insert Empty Drop	1	MyISAM	latin1_swedish_ci	1.1 KiB	-
<input type="checkbox"/> text2	Browse Structure Search Insert Empty Drop	1	MyISAM	latin1_swedish_ci	1.1 KiB	-
<input type="checkbox"/> text3	Browse Structure Search Insert Empty Drop	1	MyISAM	latin1_swedish_ci	1.1 KiB	-
<input type="checkbox"/> user	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> users	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
<b>6 tables</b>	<b>Sum</b>	<b>12</b>	<b>MyISAM</b>	<b>latin1_swedish_ci</b>	<b>51.3 KiB</b>	<b>0 B</b>

**Bild 6. Strukturen på tabellerna i databasen**

Verktyget som jag skapar databaserna med är PhpMyAdmin där man lätt kan konfigurera tabeller och lägga till flera fält i varje tabell. När man skapar en tabell så anger man hur många fält som skall finnas i tabellen. Man väljer också vilken typ av information det skall vara i varje specifikt fält. Man kan också kryptera informationen direkt i varje fält. Detta görs för att man lättare skall ha kontroll på tabellerna och underlätta när man lägger in information senare.

PhpMyAdmin körs direkt från mitt webbhotell. Jag har dock gjort det möjligt att jag direkt från min webbplats kan ändra texten i tabellerna utan att varje gång behöva logga in på webbhotellet.

För att hämta in informationen från databasen till webbsidan så används PHP enligt Kod 3. Först måste du göra en databasanslutning med `mysql_connect` i php där du anger lösenord och användarnamn till databasen så att PHP vet vilken databas och tabell som skall hämtas in.

### **Kod 3. Anslutning till databas**

```
<?php
$con = mysql_connect ("localhost", "database", "password");
if (!$con)
{
    die ('Could not connect: ' . mysql_error());
}
mysql_select_db ("database", $con);
$result = mysql_query ("SELECT * FROM text");
?>
```



När databasanslutningen är öppen och kontrollerad så hämtar man in texten från en tabell i databasen att synas på sidan enligt kod 4. Med syntaxen echo i php skriver man på skärmen ut text i php, här i kod 4 skrivs rad text1 från tabell text som hämtats från kod 3 ut.

#### ***Kod 4. Visar information från databasen***

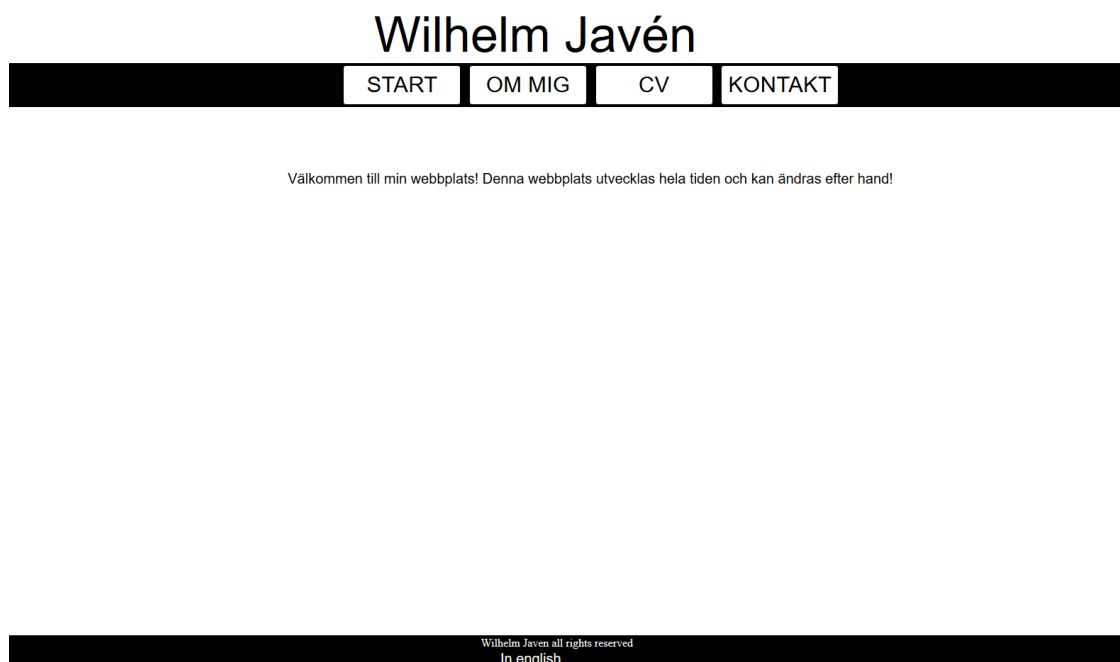
```
<?php
while ($row = mysql_fetch_array ($result))
{
    echo $row["text1"] }
?>
```

## **5. Webbsidans design**

### **5.1. Val av layout**

Jag har valt en enkel layout utan en massa extra färger eller bilder. Efter moget övervägande valde jag svart och vitt som färgtema. Det beror på att jag personligen gillar just svart och eftersom detta är en personlig sida så kör jag med de färger som jag själv tycker om.

Jag hade först en svart bakgrund men då detta är så dystert valde jag att ha vit bakgrund med svart meny balk och svart text. Menyerna ändrar när man för musen över dem (Bild 7).



**Bild 7. Startsidan index.php**

Hela webbsidan består av fyra olika sidor, Startsidan, infosidan, cv och en kontaktsida. Startsidan är den första sidan som visas när man kommer in på min webbsida. Startsidan är en introduktion till webbplatsen. På cv sidan finns min cv med meritförteckning då det är meningen att denna sida skall marknadsföra mig och visa mina erfarenheter på ett positivt sätt. CV har jag i en tabell för att få den strukturerad. På kontaktsidan finns ett formulär som besökaren kan fylla i för att få kontakt.

Webbplatsen är uppbyggd på ett ganska traditionellt sätt d.v.s. header högst uppe och navigationsmenyn under. I mitten av sidan finns själva innehållet och en footer längst ner. Vilket verkar vara mest logiskt. Jag valde detta för att det både är åskådligt och lätt att bygga upp en sida på detta vis och lätt därför också att läsa sidan. Hela webbplatsen har samma uppbyggnad. Det är endast innehållet på sidorna som kommer att variera.

När det gäller att följa idealen för grafisk, informations och interaktionsdesign har jag försökt implementera de tre så bra som möjligt. Informationsdesignen på webbplatsen är all text som jag har. Texten skall vara lättläst med en lämplig font men inte heller för mycket text på samma sida. För interaktionsdelen har jag ett formulär som man kan fylla i och trycka på för att skicka information. Tillsammans med formuläret används också en Captcha (se kapitel 6.3 för mera om detta). Menyerna är ju också en form av interaktion då man navigerar sig på webbplatsen med den. Sidan skall också vara lätt att navigera. Man brukar tala om treklicksregeln, att allting på webbplatsen skall gå att komma åt med max tre musklick.

## 5.2 Responsivitet

Responsivitet är ett relativt nytt begrepp inom webbutveckling då det handlar om att skapa webbsidor som ändrar utseende beroende på upplösningen på skärmen. Tidigare när alla använde internet från datorer med ungefär samma storlek på skärmen följde man en viss upplösning som standard, men då det idag finns så många olika enheter med olika skärmupplösningar måste man utveckla webbsidorna så att de ser så lika ut som möjligt oberoende av skärmupplösning.

För att få detta att fungera finns en del olika knep. De flesta verktyg som man utvecklar detta med brukar kunna skapa responsiva sidor ganska lätt.

Det finns också färdiga stilmallar som är helt responsiva. Tack vare css3 kan man ganska enkelt skapa flexibla webbsidor som automatiskt ändrar utseende.

Jag har gjort min webbplats så att den automatiskt skall centreras i mitten oavsett storlek på skärmen. När upplösningen på skärmen blir mindre om man surfar från en mobil eller liknande så ändrar sidan automatiskt utseende (Kod 5).

```
Kod 5. Exempel på responsivitet i css3  
@media screen and (max-width: 700px) {  
  #header{  
    font:0.5em arial;  
  }  
}
```

Detta betyder att om upplösningen på skärmen är 700 pixlar eller lägre så ändrar sidan automatiskt till det som man har satt in i koden. I detta exempel så ändras fontstorleken på hederns på sidan (Kod 5).

Syftet med responsiv design är att användarupplevelsen skall vara lika hög oavsett om du surfar från en mobil, en surfplatta eller en dator. Det kan vara utmanande att få hela sidan att fungera men man strävar efter att så många olika element på sidan som möjligt skall se bra ut. Många webbplatser idag har skilda mobilversioner, en version av sidan som är lättare och anpassad för mobiler.

## 6. Förverkligande av webbplatsen

Jag kommer nu att berätta om hur jag har byggt upp sidan och om själva processen vid uppbyggandet.

För att komma igång med det hela började jag med att skapa de HTML dokument som behövs för att göra upp strukturen på själva sidan. Därefter började jag med css att skapa de olika elementen för webbplatsen, Headern först sedan menyer och innehållet. Sist gjorde jag sidan responsiv med css så att den automatiskt anpassas till skärmstorlek.

Efter att sidan var uppe och layouten färdig så började jag med funktionerna. Databashanteringen och formuläret samt inloggningssystemet. När strukturen på sidan är uppbyggd är det lätt att i framtiden lägga till mera material och funktioner eftersom man enkelt kan placera dem på webbsidan.

## 6.1. Kodning

När det kommer till kodningen av webbplatsen har jag gjort det systematiskt. Först har jag valt ut vilken kod som behövs för det som jag bygger in på webbplatsen och sedan bestämt hur jag skall implementera koden. Jag har delvis använt mig av färdig kod från internet som jag ändrat så att den passar in med det övriga som jag har gjort. Man kan kolla upp kod från internet som man sedan implementerar på sin webbplats. Det är viktigt att förstå varför koden behövs och vad den gör för att kunna själv ändra koden så att den passar in med det man gör.

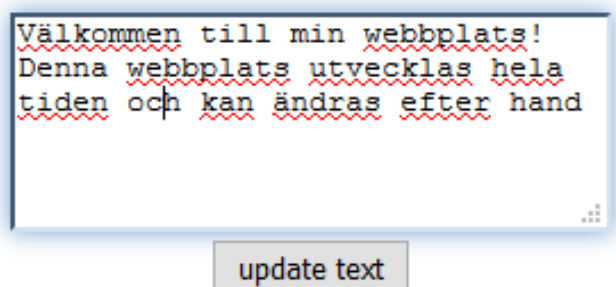
Oftast finns det flera sätt att skriva kod på så jag har hittat ett sätt som passar mig. Därför kan det vara svårt att redigera kod som någon annan har gjort eftersom man inte känner igen sig i den. PHP är ett mycket omfattande scriptspråk där det finns många olika lösningar för varje funktion. Man brukar därför skriva kommentarer i koden för att lätt kunna hitta och redigera kod.

När man kodar eller programmerar så gör man det oftast i delar d.v.s. man skapar en del i taget som fungerar förrän man går vidare och kodar nästa del. Detta för att lättare kunna hitta eventuella fel i koden som kan uppstå. Jag har först testat de script jag gjort lokalt på min dator förrän jag sedan har satt upp det på servern. Efter att det är uppe och fungerar kan jag ännu göra små ändringar då jag vet vad jag har gjort och var jag skall ändra utan att förstöra hela konceptet.

Det är också viktigt att man regelbundet tar backup på sina filer. Även om de finns på en server kan det vara bra att ha alla filer lokalt sparade om det skulle hända något med servern eller om man skulle bli utan internet.

## 6.2. Inloggning

För att göra det lättare att redigera innehållet på sidan har jag gjort ett inloggningssystem. När jag loggar in med lösenord och användarnamn så kommer det en text box fram som jag kan ändra innehållet i databasen. Inloggningen sker från en separat php fil med formulär. Texten från databasen visas i text boxen så det lätt går att ändra och sätta till text (Bild 8).



**Bild 8. Textrutan för att uppdatera databasen**

Koden för att uppdatera databasen körs i en separat php fil (Kod 6).

### ***Kod 6. Uppdatering till databas från php***

```
$sql = "UPDATE text3 SET text2='$_POST [update text]' WHERE text1='test'";

if ($conn->query ($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error; }
```

Här sätter php scriptet in i tabell 3 på raden text2 det som finns i textboxen.

Denna inloggning sker med hjälp av sessioner som kollar om användarnamnet stämmer med namnet i databasen (Kod 7).

### ***Kod 7. Kontroll av sessioner***

```
if (is_array($row)) {

$_SESSION ["id"] = $row [id];
$_SESSION ["username"] = $row [username];

} else {
$message = "Invalid Username or Password!";
}
```

Sessioner används för att spara användardata över flera filer så att man inte behöver logga in varje gång utan istället öppna en session som hålls öppen tills man loggar ut och då stängs den. Sessioner sparar tillfälligt informationen på servern och förstörs när man stänger sessionen. För att öppna en session skriver man i php `session_start()`; och stänger sessioner med `session_destroy()`; För att öka säkerheten på sidan kan man även i php ställa in sessioner så att de är öppna en viss tid och sedan stängs automatiskt. (Tutorialspot, 2015).

### 6.3. Säkerhet

Då det är viktigt att sidan hålls säker så är lösenordet i databasen krypterat. På min webbplats har jag krypterat lösenordet i databasen med md5 som är en enkel krypteringsmetod som kan avläsas av mysql. Md5 är dock ganska lätt att komma åt så man bör se till att webbplatsen är säker från så kallade Brute-force Attacker, vilket innebär att ett automatiskt program gissar olika kombinationer av lösenord. Det finns ett antal sätt att förhindra att detta kan ske.

Det första man bör se till är att man har ett långt lösenord med många olika tecken då det blir svårare att gissa för programmet desto mera tecken man använder. För att göra lösenordet mera säkert kan man förutom att kryptera det också salta det. Vilket innebär att man slumpvis lägger till extra tecken som gör att det blir svårare att lista ut det riktiga lösenordet. Orsaken till att man gör detta är att det är relativt lätt att koda om ett vanligt krypterat lösenord men om man saltar det så sätts andra tecken till krypteringen vilket gör det genast svårare att avläsa.



Man kan också i php sätta en gräns för hur många inloggningsförsök man kan göra innan inloggningen stängs. Man skall också se till att inte webbplatsen sparar användardata eller andra uppgifter i filer på servern då dessa kan komma åt. En annan metod är att använda SSL vilket betyder att hela webbplatsens information krypteras. (Owasp, 2015).

## 6.4. Formulär

När man har ett formulär på en webbsida bör man tydligt visa varför informationen skall fyllas i och syftet med det. Här har vi en form av interaktionsdesign (se kapitel 5.4). Formuläret skall vara lätt att använda så att besökaren på sidan enkelt förstår hur det fungerar.

På min kontaktsida har jag satt in ett formulär som man kan fylla i om man vill för att få reda på något eller ge feedback. De fält jag har på formuläret är: ämne, meddelande, epost och en Captcha. Fältet epost satte jag in eftersom det kan vara bra att kunna svara på den feedback som man får och då är det bra att veta vem som skickat det. Man behöver dock inte fylla i detta fält om man vill vara anonym. De övriga fälten bör däremot vara ifyllda. Formuläret skickar informationen direkt till min epost.

För att få det att fungera har jag php kod som hämtar in informationen från textrutorna och sedan skickar den till min e-post. Formuläret varnar också om man glömt att fylla i ett fält. För att få formuläret att se bra ut och hålla sig inom marginaler har jag satt det i en tabell i html, där texten och fälten är i skilda kolumner. Angående utseende har jag svarta textrutor för att hålla samma färgtema som på den övriga webbplatsen (Bild 9).

Ämne:	<input type="text" value="ämne"/>
meddelande:	<input type="text" value="skriv ditt meddelande här"/>
epost:	<input type="text" value="epost"/>
Ange texten i bilden:	<input type="text" value="hb06Fq"/>  
	<input type="button" value="skicka"/>

**Bild 9. Bild på hur formuläret ser ut**

Koden körs i en separat fil verify.php som används i samband med Captchan och kontrollerar om formuläret är korrekt ifyllt. I Kod 8 finns koden för att sända e-post.

Först kommer variablerna som innehåller namnen på textboxarna och de hämtas sedan in från en javascriptfil med post funktionen. I headers sätts sedan in diverse information om e-posten som skickas. Till sist kommer själva mail funktionen som innehåller allt det som har angivits.

### **Kod 8. Sändning av e-post**

```
<?php
$to = "epost";
$subject = $_POST ["subject"];
$message = $_POST ["textarea"];
$headers = "From: example@example.com" . "\r\n";
$headers. = 'Content-Type: text/plain; charset=utf-8';
mail ($to, $subject, $message, $headers, '-fsomeone@example.com');
?>
```

För att mail funktionen skall fungera måste man ha en SMTP(Simple Mail Transfer Protocol) mail server som används för att skicka e-post och finns som standard på de flesta webbservrar.

#### **6.4.1. CAPTCHA**

För att förhindra onödig spam som skickas från formuläret har jag satt in en så kallad CAPTCHA eller ett robotfilter på svenska. Den används för att verifiera att den som fyller i formuläret är en mänsklig person. Syftet med detta är i första hand att undvika angrepp eller att ett automatiskt program fyller i formuläret.

En CAPTCHA är ett test som anses vara lätt att lösa för människor men svårt för ett program. Captchor används idag på många olika webbsidor främst vid inloggningar för att öka säkerheten då man förutom att ange lösenord och användarnamn också måste ange rätt kombination av Captchan, vilket gör det svårare att lyckas komma in på en sida och förhindrar ett automatiskt program att logga in. Captchor kan också användas på formulär eller vid nedladdning av filer.

Jag satte in en Captcha på min webbplats beroende på att jag ville ha något mera än bara formuläret. Captchan behövs också för att förhindra att det skickas tomma meddelanden till e-posten. Captchor kan se olika ut men består oftast av en kombination av text och siffror som är lätta att avläsa. (Captcha, 2010)

Captchan som finns på min webbplats genereras av en php fil som slumpmässigt skapar en kombination av siffror och bokstäver. Verifieringen sker sedan av en javascript fil som kollar



att Captchan stämmer med det som man har skrivit in. Om det stämmer så skickar den informationen vidare till ett annat php script som sedan skickar mailet. Om informationen är felaktig så kommer det upp en varningsskärm. Formuläret är kopplat till Captchan så att det inte skickar någonting om formuläret är tomt.

För att kontrollera att formuläret är ifyllt så skapar man en If sats i javascript som kollar om formuläret är ifyllt (Kod 9). Först identifierar man de olika textrutorna i formuläret med namnen de har: subject, textarea och captcha1. Sedan anger man om de är tomma eller inte. Om de är tomma så kommer det fram en varningsskärm som meddelar att alla fält skall ifyllas. Om de inte är tomma så går scriptet vidare till nästa fas.

### ***Kod 9. Kontroll av formulär***

```
//validation function
$('#submit').click(function () {
var subject = $("#subject").val();
var message = $("#textarea").val();
var captcha = $("#captcha1").val();

if (subject == '' || message == '' || captcha == '')
{
alert("fyll Alla fält");
}
}
```

För att skapa själva Captchan så används följande phpscript (Kod 10). Här skapas variabler med tecken som Captchan sedan använder. Bakgrunden i Captchan hämtas från en bild bj.jpg.

Scriptet öppnar också en session då informationen från kod 10 kommer att användas i flera script. Detta script sätts in i tabellen med formuläret så att Captchan syns som en bild. Captchan används sedan tillsammans med kod 8 som sänder eposten bara om Captchan är korrekt ifyllt.

### ***Kod 10. Generering av Captcha***

```
<?php
session_start();
$captchanumber = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyz';
$captchanumber = substr (str_shuffle ($captchanumber), 0, 6);

$_SESSION ["code"] = $captchanumber;

$image = imagecreatefromjpeg ("bj.jpg");
$foreground = imagecolorallocate ($image, 175, 199, 200); //font color
imagestring ($image, 5, 45, 8, $captchanumber, $foreground);
header ('Content-type: image/png');

imagepng($image); ?>
```

Bredvid Captchan finns en knapp som genererar en ny slumpad Captcha om man vill byta till en annan utan att uppdatera sidan (Kod 11).

### ***Kod 11. Uppdatering av Captcha***

```
$("#reload").click(function() {
$("#img#img").remove();
var id = Math.random();
$('').appendTo("#imgdiv");
id = '';
});
```

Bilden som man trycker på har namnet reload för att identifiera knappen. Sedan skapas en funktion som ger en slumpmässig kombination av siffror och bokstäver som hämtas från captcha.php. Efter det sätts den nyskapade kombinationen in på Captchans plats som är imgdiv och som finns i tabellen med formuläret i html filen.

Vanligtvis så sänder man information från ett formulär till php med post metoden men då verifieringen sker i en JavaScript fil så används Ajax för att skicka informationen vidare från JavaScript filen till verify.php filen. I kod 12 finns koden för detta som berättar att informationen från text boxarna skickas vidare till verify.php med post metoden.

Ajax (Asynchronous JavaScript and XML) är en samling olika tekniker som används för att bygga webbapplikationer med bättre interaktivitet jämfört med andra metoder. Ajax är också lättare att använda än tidigare tekniker. Med Ajax kan man bl.a. skapa innehåll som uppdateras utan att man måste ladda om hela webbsidan. En text kan uppdateras automatiskt utan att användaren måste uppdatera sidan. Ajax laddas också in betydligt snabbare än äldre tekniker. (W3schools, u.å.).

I Kod 12 används Ajax för att automatiskt sända data vidare till verify.php efter att en kontroll utförts tidigare i scriptet.

### **Kod 12. Ajax sändning av data till verify.php**

```
$.ajax({
type: "POST",
url: "../verify.php",
data: {captcha: captcha, message: message, subject: subject},
success: function (html) {
alert (html);}
```

## 6.5. Språk

Från början hade jag tänkt att webbplatsen skulle vara på svenska men då det är bra att flera människor kan läsa min webbplats så bestämde jag mig för att också göra den på engelska. För att få den på två språk sätter jag in en länk i footern som byter språk.

Jag har skapat två versioner av varje fil, en svensk och en engelsk. Jag måste även ha en tabell för varje språk i databasen eftersom texten hämtas därifrån. Startsidan kommer att vara på svenska och vill man sedan ha hela webbplatsen på engelska så trycker man på länken i footern.

## 7. Slutsats

När slutligen min webbplats är uppe på adressen wilhelmjaven.com och fungerar kan man fundera över vad som kunde ha gjorts bättre och vad som är bra. Jag är nöjd med layouten på sidan då jag tycker det är utmanande att skapa bra layout som ser fin ut. Efterhand så kan jag tycka att jag kunde ha gjort webbplatsen på ett annat sätt och ändra utseende lite.

De funktioner som jag har byggt in fungerar bra och jag tycker det går smidigt att använda webbplatsen. Jag har också fått positiv feedback från mina kompisar och de som jag bett testa min sida. Det har också varit utmanande att få en del funktioner att fungera såsom inloggningen. Jag undersökte olika möjligheter innan jag fick klart för mig hur jag skulle göra den. Det fodras att man sällar ut den mest relevanta informationen från internet. Planeringen av informationssökandet är viktigt för att underlätta arbetet.

Jag inser att man ska planera arbetet väl då är det lättare att efter varje steg gå vidare. Jag kunde eventuellt ha använt mig av något verktyg eller på något sätt kombinerat arbetet med flera metoder. Det är inte alltid det lättaste att lyckas skapa en bra webbplats från grunden och då kunde jag eventuellt ha använt mig av någon annan metod för att få en bättre helhet.

Jag kommer att uppdatera och fortsätta utveckla webbplatsen i framtiden vartefter jag kommer på material som jag vill ha med. Det kommer även att gå lättare för mig i fortsättningen att utveckla webbplatser då jag nu har fått erfarenhet och vet vad som krävs.

Svårigheten med att utveckla en webbplats åt sig själv genom att koda är att det tar mera tid att komma igång och att lyckas få webbplatsen att fungera som man vill. Naturligtvis är det lättare att använda sig av de metoder som finns idag då de innehåller ett färdigt gränssnitt som man kan bygga vidare på och kunna genom dem lära sig att utveckla professionella webbsidor. Jag anser dock att jag lärt mig otroligt mycket genom att göra allt från grunden, så det kan jag rekommendera åt andra.

Fortfarande finns det en hel del som kan förbättras på min webbplats men jag har skapat en fungerande grund som går lätt att nu bygga vidare på. Det har varit mycket givande att göra detta arbete och jag är i det hela mycket nöjd med slutresultatet.

## 8. Sammanfattning

I detta examensarbete har utvecklandet av en fungerande webbplats behandlats både teoretiskt och praktiskt. Det har också behandlats teorier om vilka olika faktorer som behövs inom webbutveckling. Man kan konstatera att det idag finns ett flertal olika sätt och metoder att skapa webbsidor på. Jag har behandlat CMS systemet Joomla och hur man använder det samt berört dess funktionalitet och struktur i mitt koncept.

Fokus har varit på skapandet av ett enkelt skal för en webbplats från grunden genom att använda kod och göra det mesta själv. Det krävs mera kunskaper och tid att sätta sig in i allt, men man lär sig mera genom att själv ta reda på informationen och bygga upp från grunden. Jag har också beskrivit den kod som finns på min webbplats så att man bättre skall förstå hur webbplatsen är uppbyggd och vad den består av.

Användbarheten måste beaktas vid utvecklingen av webbsidor. För att kunna skapa webbsidor måste man förstå och kunna använda de viktigaste faktorerna inom webbutveckling.

## Källförteckning

**Apachefriends, About. (u.å.)**

<https://www.apachefriends.org/about.html>[hämtad 14.9.2015]

**Bacchus, Css navigation bar. Publicerad 13.5.2013**

<http://stackoverflow.com/questions/5995405/how-to-center-navigation-bar-with-css-or-html>  
[hämtad 25.6.2015]

**Bebabatteronline, Grundhandbok om sökmotoroptimering. (u.å.)**

<https://www.bebetteronline.com/wp-content/uploads/2014/12/search-engine-optimization-starter-guide-sv.pdf> [hämtad 9.10.2015]

**Brasil, P., Understanding the Joomla directory structure. Publicerad 30.4.2012**

<http://www.grimmster.com/development/understanding-the-joomla-directory-structure>  
[hämtad 29.10.2015]

**Capcha, captcha: Telling Humans and Computers Apart Automatically. (u.å.)**

<http://www.captcha.net/> [hämtad 10.9.2015]

**Digicert, What is ssl? (u.å.)**

<https://www.digicert.com/ssl.htm> [hämtad 13.9.2015]

**Formget, captcha code in php and jquery. (u.å.)**

<http://www.formget.com/captcha-code-php-jquery/> [hämtad 28.8.2015]

**Goodwin, R., What is css3? Publicerad 10.9.2013**

<http://www.sitepoint.com/web-foundations/what-is-css3/> [hämtad 18.9.2015]

**Jeffknupp, What is a Web framework? Publicerad 3.3.2014**

<https://www.jeffknupp.com/blog/2014/03/03/what-is-a-web-framework/> [hämtad 28.9.2015]

**Joomla, About Joomla (u.å.)**

<https://www.joomla.org/about-joomla.html> [hämtad 4.10.2015]

**McCarthy-Parton, T, Introduktion till frågespråket SQL. Uppdaterad 18.7.2005**

<http://www.databasteknik.se/webbkursen/sql> [hämtad 2.9.2015]

**Owasp, Blocking Brute Force Attacks. Uppdaterad 18.3.2015**

[https://www.owasp.org/index.php/Blocking\\_Brute\\_Force\\_Attacks](https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks) [hämtad 2.11.2015]

**Phpeveryday, Backend Structure [online]**

[www.phpeveryday.com/articles/3-1-Back-End-Structure-P639.html](http://www.phpeveryday.com/articles/3-1-Back-End-Structure-P639.html)[hämtad 11.10.2015]

**Rocheleau, J., Responsive mobile layout (u.å.)**  
<http://www.hongkiat.com/blog/responsive-for-mobile-screens/> [hämtad 5.7.2015]

**Rohde, M., Crash Course: Create a Joomla! Template from Scratch. (u.å.)**  
<http://www.htmlgoodies.com/beyond/webmaster/toolbox/crash-course-create-a-joomla-template-from-scratch.html> [hämtad 30.10.2015]

**Susnet, Översikt av databaser. (u.å.)**  
<http://susnet.nu/databaser.html> [hämtad 2.9.2015]

**Söderström, J., En sajt kräver tre sorters design. Publicerad 8.11.2001**  
<http://kornet.nu/3xdesign.shtml> [hämtad 25.8.2015]

**Tutorialspot, PHP Sessions. (u.å.)**  
[http://www.tutorialspoint.com/php/php\\_sessions.htm](http://www.tutorialspoint.com/php/php_sessions.htm) [hämtad 23.9.2015]

**Vincy, PHP login script with session. Publicerad 22.7.2013**  
<http://phppot.com/php/php-login-script-with-session/> [hämtad 10.9.2015]

**W3schools, AJAX Introduction. (u.å.)**  
[http://www.w3schools.com/Ajax/ajax\\_intro.asp](http://www.w3schools.com/Ajax/ajax_intro.asp) [hämtad 31.10.2015]

**Webbdesignskolan, Information om Css. (u.å.)**  
<http://webbdesignskolan.se/css/css.php#info> [hämtad 20.7.2015]

**Webbdesignguiden, Webbdesign på svenska. (u.å.)**  
<http://www.webbdesignguiden.se/> [hämtad 12.8.2015]

