

# Rapport på uppgift 3 i webbprogrammering

Kurskod: 1ME205, Datum 2015-08-21  
Webbadress: <http://1me205.lnu.se/~rk222ed/u3/>

av Robin Kanthe (810519-4917)

## 1. Konstruktion av applikationen

Applikationen är skriven uteslutande med html, js och php och består av följande sidor:

- login.php - där alla användare loggar in. Här kan man också registrera sig som ny användare. Jag har gjort två användare av presentationssyfte: 1. Email: adam@adam Password: adamadam 2. Email: amanda@amanda Password: amandaamanda Man kan inte komma in på någon sida utan att vara inloggad. All data hämtas genom formulär eller AJAX-anrop till ett php-skript som hämtar och skickar tillbaks med anrop till en MySQL-server där all data ligger lagrad. Därför kan ingen användare komma åt datan direkt. Åtgärder har också vidtagits för att undvika att hackare skickar skadliga skript genom formulären. Formulärsfälts text omvandlas till vanlig text, så att inte html-koder kan skrivas in och t ex skicka skadliga skript. Skript mot sk. email injects, alltså ett sätt att skicka spam-email genom formulär motverkas av ett php-skript (*Powers 2010, s118-121*). Åtgärd mot att andra serverar skickar mängder med data genom formulären, utan att skriva in i dem används. En krypterad kod skapas och sparas på både serversidan och klientsidan. Den skickas gömd med formuläret och jämförs. Då vet man att datan skickats genom formuläret. (*CSS Tricks (2009)*). Lösenord krypteras med php-funktionen *crypt*. När användare läggs till, kontrolleras uppgifterna på samma sätt, samt att e-mail och username inte redan finns, att lösenordet, som skrivs två gånger, är identiskt på båda och 8 tecken eller fler långt.
- index.php – eller Kweets, alternativt MyKweets i navigeringen. I Kweets kan man se alla användares statusuppdateringar. I MyKweets syns bara ens egna uppdateringar och ens namn hämtas i databasen med AJAX-anrop och php-skript. På det sättet hämtas även all info om uppdateringarna. De görs till objekt i Kweet.js och presenteras på index.php med Home.js. Egna uppdateringar får en "Delete"-knapp som tar bort aktuellt Kweet-objekt. Det finns då inte längre med i databasen eller syns på skärmen. I index.php görs även nya statusuppdateringar och det kontrolleras att texten har under 140 tecken.
- contacts.php – Här hämtas all info om användarna och presenteras med Contacts.js. Varje person läggs i ett Contact.js-objekt. Med en knapp i objekten kan man välja att följa/inte följa en användare. Då ändras följarsstatus i databasen via php-skript. Med länkar som aktiveras med AJAX-anrop hämtas alla personer, personer man följer eller personer som följer en.
- settings.php – En bild kan laddas upp från hårddisk med php-skript via ett formulär. (Bestående av en knapp.) Kontroller av storlek och filtyp görs och av att filen skickas på rätt sätt. Funktionen som kontrollerar filen är skrivet till stor del baserat på ett föreläsningsexempel. I Settings kan man även uppdatera ens användaruppgifter via ett formulär. Formuläret skickas till php-skript, som kontrollerar att användarnamn inte redan finns, att allt finns med och att lösenordet skrivs in rätt (identiskt lösenord är inskrivet 2 gånger och att det är 8 tecken eller fler långt). Detta görs likadant i AddUser och lösenord krypteras givetvis.

Delar av sidorna index.php, contacts.php och settings.php, samt flera skript som är gemensamma, inkluderas till dem i filen header.php. Till sin hjälp har alla sidor js och php-skript. Alla skript har

grundläggande felhantering med try-catch-satser. När man försöker logga in sparas skickad data i sessionsvariabler som sedan automatiskt skrivs in i formuläret igen om användare råkat skriva fel. Tydliga felmeddelanden skickas också med sessionsvariabler och presenteras. Vid godkänd inloggning, sparas användar-id och krypterat lösenord i sessionsvariabler. De används sen för att kontrollera användaren vid varje sidladdning. Vid utloggning tas alla sessions-variabler bort.

## 2. En egen reflektion om skillnaden mellan att använda XML och MySQL som databas

MySQL (liksom det grafiska verktyget phpMyAdmin) och XML är gratis, plattformsoberoende och dominerande när det gäller att lagra data för webbplatser (*Powers 2010, s.280*). Det finns mycket information/hjälp/tips om MySQL och XML i böcker, på webbplatser och i forum. MySQL och XML lagrar data på lite olika sätt och har sina fördelar och nackdelar.

### Fördelar med MySQL

- Öppen källkod: Programmerare kan modifiera MySQL för sina egna behov. (*Powers 2010, s.280*)
- Är kompatibel med många programspråk t ex PHP, PERL, C, C++ och JAVA (*Tutorialspoint – MySQL Introduction (2011)*).
- Stöder flera datatyper, t ex INT, VARCHAR och TEXT. (*Powers 2010, s.299*)
- Kan lagra datum och tid och kan göra "Timestamps" när man sätter in/uppdaterar i databasen. (*Powers 2010, s.300*)
- MySQL är designad att hämta information snabbt (*Hubpages - Advantages and Disadvantages of MySQL (2015)*).
- Det går inte att komma åt informationen utan access till databasen som alltid skyddas med krypterade lösenord. (*Hubpages - Advantages and Disadvantages of MySQL (2015)*)
- Felhantering, automatisk backup och återställning. (*Hubpages - Advantages and Disadvantages of MySQL (2015)*)
- Relations-struktur: All data struktureras i entiteter ("objekt") och relationer mellan dessa. Detta sätt är mycket intuitivt och dessutom är objektorienterade programmerare vana vid detta. (*Powers 2010, s.281*)
- Du kan använda GUI program som phpMyAdmin för att se struktur och data visuellt som tabeller och manipulera datan på ett enkelt sätt med olika verktyg (*Powers 2010, s.284*).

### Nackdelar med MySQL

- Servern måste ha installerat MySQL, Apache och ett programspråk, t ex PHP, för att göra MySQL sökningar. Det är också fördelaktigt att ha ett GUI som phpMyAdmin installerad.
- Om anslutningen till databasen upphör förlorar man åtkomst till all lagrad data. Eftersom XML är filer på servern krävs att man förlorar kontakten med servern för att detta skall hända. Och då har man inte någon nytta av informationen eftersom hela webbplatsen då är nere.

### Fördelar med XML (*TechMynd - Advantages & Disadvantages of XML (2015)*)

- Kan läsas av både människa och dator.
- Stöder Unicode, dvs kan tyda nästan alla skriftspråk.

- Internationell standard.
- Plattformsberoende

## Nackdelar med XML

- Stora databaser gjorda i XML kan bli svårlästa för människor då ett XML-dokument är en lång kedja av taggar och sluttaggar.
- Till skillnad från MySQL hämtas data i XML filer som är textbaserade. Att hämta textbaserade filer via filsystemet går relativt långsamt. (*InformIT – XML Performance and Size*)
- Med debugverktyg går det att komma åt känslig information som lösenord.
- Inga datatyper (*TechMynd - Advantages & Disadvantages of XML (2015)*).
- Hierarkisk struktur: Begränsad jämfört med relations-struktur. (*TechMynd - Advantages & Disadvantages of XML (2015)*)

Min slutsats av detta är att bästa sättet att använda MySQL och XML är att använda MySQL för att säkert lagra information i den egna databasen för att snabbt och säkert kunna plocka fram den för användning. XML kan användas för att skicka mindre känslig information till webbplatser/applikationer utanför den egna webbplatsen. Man kan inte gärna låta andra få tillgång hela ens databas bara för att de ska få tillgång till viss information från den. Även icke webbaserade applikationer kan programmeras för att läsa XML-filer.

## 3. Referenser

David Powers (2010), *PHP Solutions – Dynamic Web Design Made Easy, Second edition*, Apress, ISBN: 978-1-4302-3249-0

CSS Tricks – Serious Form Security <<https://css-tricks.com/serious-form-security>> [2009-05-19]

Tutorialspoint – MySQL Introduction <<http://www.tutorialspoint.com/mysql/mysql-introduction.htm>> [2015-08-12]

Hubpages - Advantages and Disadvantages of MySQL  
<<http://gautambikash.hubpages.com/hub/Advantages-and-Disadvantages-of-MySQL#>> [2015-08-15]

TechMynd - Advantages & Disadvantages of XML <<http://www.techmynd.com/advantages-disadvantages-of-xml/>> [2015-02-11]

InformIT – XML Performance and Size <<http://www.informit.com/articles/article.aspx?p=367637&seqNum=2>> [2015-08-15]