

## Webbteknik II Projektrapport

### Inledning

Mitt projekt har handlat om att bygga en musiksida som sammanfogar information om ett band/en artist från olika källor och sammanfattar detta på en sida. Dom API'er som jag använder för närvarande är Last.FM och Wikipedia. Sidan är tänkt att användas som en plats där besökare kan hitta mer information om sina favoritartister. Det är inte meningen att man ska upptäcka ny musik via sidan, utan snarare kunna hitta mer information om sina favoritband.

En liknande applikation jag hittade var [www.tastekid.com](http://www.tastekid.com). Denna sida tar användarens sökning och hittar andra artister som är liknande det man sökte efter. Tastekid är dock inte bara för musik, utan man kan också söka på spel, böcker, och annat.

### Teknik

Den teknik jag har använt är följande:

- HTML
- CSS
- Javascript/jQuery
- Ajax
- PHP
- MySQL

### Serversida

På serversidan sköter jag anropen till API'erna med hjälp av cURL. Jag har i mångt och mycket försökt dela upp funktionaliteten mellan olika php-filer för att skapa en bra överblick och för att gynna utbyggbarheten av applikationen. Separata lager på serversidan hanterar olika saker så som validering av input, användarregistrering, användarlogin, databasanslutning, och sökningar.

## Klientsida

På klientsidan har jag delat upp hanteringen i tre beståndsdelar: användarhantering, sökningar, och allting som behöver hanteras när sidan laddas. Jag har fortsatt med tidigare filosofi att försöka skapa en så bra grund för utbyggnad som möjligt. Input från användaren hanteras via javascript och information skickas vidare till serversidan via asynkrona ajax-anrop.

## Säkerhet och prestandaoptimering

För att se vilket state sidan befinner sig i använder jag asynkrona Ajax-anrop, till exempel för att se om någon användare är inloggad eller inte, och därigenom vilken information som ska visas på sidan. Då jag endast har en bild med på sidan, sidans logotyp, laddar jag denna endast en gång i sidans index-fil. Detta gör att laddningen av sidan går snabbare då bilden inte laddas om och om igen via ajax-anropen. Övrig kod så som inloggningsformulär hämtas från en PHP-fil och läggs in på korrekt ställe på sidan. Detta gör att väldigt lite htmlkod behöver laddas för att användaren ska se att sidan har laddat.

Angående säkerheten finns det lite kvar att önska. Just nu visas inte lösenorden i klartext, men det är allt man kan säga om lösenordssäkerheten. I databasen sparas fortfarande lösenorden i klartext. För att skydda användarna vill jag tillämpa någon slags hashing av lösenorden, att föredra är "salted hashing". Ytterligare säkerhetsåtgärder kommer handla om att skydda användaren mot sessionskapning, till exempel genom att kopiera sessionskakan till andra webbläsare.

Sidan har säkerhetsåtgärder mot inskjutning av script, så att ingen användare ska kunna lägga in script som hindrar sidans funktionalitet. Detta hanteras via validering som ser till att ingen input kan innehålla taggar.

Sedan för att ytterligare öka prestandan på sidan vill jag lägga till så att en sökning som redan är presenterad för användaren inte laddas om på nytt om användaren skickar samma sökord. Just nu finns också en laddnings-gif som visas medans användarens sökning är under behandling. Jag har även sett till att hindra formuläret från att skickas upprepade gånger under tiden en sökning pågår, detta för att hindra användaren från att skicka massor av sökningar under kort tid och på så sätt överbelasta API'erna.

## Offline-first

Dom tankar jag haft angående offline-first har varit att det inte var prioriterat för sidan, då det först och främst krävs en inloggning för att kunna använda applikationen, och har inte

användaren någon anslutning kommer dom inte kunna valideras varken mot den inbyggda användarhanteringen eller via Facebook.

Det jag anser kunna vara bra att implementera i framtiden är en offlinefunktion när man gör en sökning på sidan. Om användaren tappar anslutning mot nätet men trots detta försöker göra en sökning skulle sidan då kunna upplysa om att anslutningen gick förlorad men också att sökningen sparats och kommer utföras så fort en anslutning upprättas igen. Detta är då inte implementerat i dagens iteration av applikationen, men står högt i prioritet för framtiden.

## **Egna reflektioner**

Jag har under det här projektet fått några nya insikter när det kommer till att skapa en sida. Först och främst insåg jag att en inloggningsfunktion via Facebook absolut bör utvecklas parallellt med eventuella andra inloggningssystem. I detta projekt skapade jag ett eget användarsystem först och arbetade länge utifrån detta, och därför blev det onödigt komplicerat för mig att implementera Facebooks inloggning i efterhand. När jag tänker tillbaka hade det nog varit bättre att enbart använda Facebook-inloggningen, och göra inloggningen för användaren valfri, men med mer funktioner som inloggad användare.

Sedan är det viss funktionalitet som jag inte hann med att lägga in då jag prioriterade andra delar av sidan. Till exempel kan man inte spara några sökningar på sin användarprofil för tillfället, och någon autocomplete-funktion finns för närvarande inte på sökfältet. Detta är dock saker jag hoppas kunna lägga till i framtiden. Jag känner att jag har skapat en applikation som jag enkelt kommer kunna bygga ut längre fram.

Jag är nöjd med applikationen i dagsläget, men ser även fram emot att fortsätta jobba på den då jag tycker det finns potential för att erbjuda mycket mer till användaren.

## **Risker**

Dom risker som finns för närvarande är främst vissa säkerhetshål som kan motverkas. Det som är viktigast att lösa just nu är i mitt tycke sessionssäkerheten. För närvarande skulle någon kunnig person eventuellt kunna ta över sessionen. Det finns inte så mycket en sån person skulle kunna komma över, men då jag planerat att användare ska kunna spara information på sina användarprofiler i framtiden så vill jag självklart inte att någon ska kunna utnyttja sidan och sessionen på något sätt.