Laboration 1 – jQuery och AJAX

I denna laboration skall ni utgå från en ofärdig webbapplikation, “The MusicSite”, och lägga till funktionalitet med hjälp av jQuery och AJAX. Utgångspunkten är att genom ”The MusicSite” skall en administratör (efter att ha validerats som giltig admin - mer om detta i laboration 2) kunna administrera artister (adminArtist.php), låtar (adminSong.php) och kommentarer (adminComment.php). Därtill skall en vanlig användare (utan validering – mer om detta i laboration 2) kunna söka bland dessa artister och låtar (search.php) samt kommentera utsökta låtar. I denna första laboration skall ni fokusera på jQuery och AJAX samt att med jQuery UI Accordion och jQuery Plugin SlimBox 2 lägga till mer funktionalitet. Innan ni börjar med att lösa laborationen skall ni ”packa upp” musicsite.zip (se it’s Learning) och spara det i den katalog som Apache ”pekar på” (www för WAMP och htdocs för XAMPP). Därefter skall ni på valfritt ställe leta rätt på två jpg bilder som ni sparar i upload_jpg som acdc.jpg och laleh.jpg. Därtill skall ni leta rätt på två ogg ljud som ni sparar i upload_ogg som wheels.ogg och colors.ogg. Därefter skall ni bekanta er med projektet genom att titta igenom vilka filer som finns med, vad filerna innehåller och avslutningsvis ändra så att sidan visas som ni önskar. Genom att studera och ändra CSS:en kommer ni att skapa er en förståelse för hur webbapplikationen är uppbyggd och hur ni skall manipulera dess struktur.

Från beställaren finns följande generella krav på er lösning

- Er lösning skall utgå från den ofärdiga webbapplikationen som finns på it’s Learning (musicsite.zip).
- Er lösning skall använda jQuery för alla moment (undantaget är jQuery UI Accordion och jQuery Plugin SlimBox 2 där HTML5 får användas).
- Er lösning skall vara unobtrusive.
- Er lösning skall fungera felfritt i senaste versionen av Mozilla Firefox.
- Er lösning skall fungera så som demonstreras i lab1_musicsite.mp4.

Från beställaren finns följande krav för adminComment

- När användaren trycker på ”Delete” skall hon/han få en bekräftelsefråga (OK/Cancel) om hon/han verkligen vill ta bort posten. Om användaren trycker på ”OK” skall formulärdata skickas (submittas) till servern. Om användaren istället trycker på ”Cancel” skall default-beteendet (en submit) avbrytas och likaså att klickhändelsen ”bubblar” till elementets förälder. Ni skall för detta krav använda er av den färdiga funktionen verifyDeleteOfComment (se script/commentFunctions.js).
- Samtliga kommentarer skall visas i en jQuery UI Accordion.

Från beställaren finns följande krav för adminSong

- När användaren trycker på ”Delete” skall hon/han få en bekräftelsefråga (OK/Cancel) om hon/han verkligen vill ta bort posten. Om användaren trycker på ”OK” skall formulärdata skickas (submittas) till servern. Om användaren istället trycker på ”Cancel” skall default-beteendet (en submit) avbrytas och likaså att klickhändelsen ”bubblar” till elementets förälder. Ni skall för detta krav använda er av den färdiga funktionen verifyDeleteOfSong (se script/songFunctions.js).
- När användaren trycker på ”Edit” skall data från aktuellt formulär kopieras till det övre formuläret vilket möjliggör att användaren kan redigera en post. Ni skall för detta krav använda er av den färdiga funktionen copySongFormData (se script/songFunctions.js).

- När användaren trycker på ”Reset” skall formulärdata rensas. Ni skall för detta krav använda er av den färdiga funktionen resetSongFormData (se script/songFunctions.js).
- När användaren trycker på ”Save” skall formulärdata valideras och om inmatad data uppfyller givna villkor skall formulärdata skickas (submittas) till servern. Om däremot inte data uppfyller givna villkor skall default-beteendet (en submit) avbrytas och likaså att klickhändelsen ”bubblar” till elementets förälder. Ni skall för detta krav använda er av den färdiga funktionen validateSongFormData (se script/songFunctions.js).
- Samtliga sånger skall visas i en jQuery UI Accordion.

Från beställaren finns följande krav för adminArtist

- När användaren trycker på ”Delete” skall hon/han få en bekräftelsefråga (OK/Cancel) om hon/han verkligen vill ta bort posten. Om användaren trycker på ”OK” skall formulärdata skickas (submittas) till servern. Om användaren istället trycker på ”Cancel” skall default-beteendet (en submit) avbrytas och likaså att klickhändelsen ”bubblar” till elementets förälder. Ni skall för detta krav använda er av den färdiga funktionen verifyDeleteOfArtist (se script/artistFunctions.js).
- När användaren trycker på ”Edit” skall data från aktuellt formulär kopieras till det övre formuläret vilket möjliggör att användaren kan redigera en post. Ni skall för detta krav använda er av den färdiga funktionen copyArtistFormData (se script/artistFunctions.js).
- När användaren trycker på ”Reset” skall formulärdata rensas. Ni skall för detta krav använda er av den färdiga funktionen resetArtistFormData (se script/artistFunctions.js).
- När användaren trycker på ”Save” skall formulärdata valideras och om inmatad data uppfyller givna villkor skall formulärdata skickas (submittas) till servern. Om däremot inte data uppfyller givna villkor skall default-beteendet (en submit) avbrytas och likaså att klickhändelsen ”bubblar” till elementets förälder. Ni skall för detta krav använda er av den färdiga funktionen validateArtistFormData (se script/artistFunctions.js).
- Samtliga artister skall visas i en jQuery UI Accordion.

Från beställaren finns följande krav för search.php

- När användaren ser den färdigladdade sidan skall hon/han se tre länkar för respektive framsökt sång: ”Show all comments”, ”Comment [filnamn.ogg]” och ”Like [filnamn.ogg]”. Existerande kommentarer och formuläret för att lägga till en ny kommentar skall alltså inte vara synliga.
- När användaren klickar på länken ”Show all comments” och inga kommentarer är synliga skall samtliga kommentarer göras synliga med någon form av animering.
- När användaren klickar på länken ”Show all comments” och samtliga kommentarer är synliga skall dessa döljas med någon form av animering.
- Givetvis skall också default-beteendet vid klick på ”Show all comments” avbrytas och likaså att klickhändelsen ”bubblar” till elementets förälder.
- När användaren klickar på länken ”Comment [filnamn.ogg]” och formuläret inte är synligt skall formuläret visas med någon form av animering.
- När användaren klickar på länken ” Comment [filnamn.ogg]” och formuläret är synligt skall detta döljas med någon form av animering.
- Givetvis skall också default-beteendet vid klick på ”Comment [filnamn.ogg]” avbrytas och likaså att klickhändelsen ”bubblar” till elementets förälder.

- När formuläret för att kommentera en sång är synligt och användaren klickar på ”Save” skall default-beteendet för ett formulär avbrytas och likaså att klickhändelsen ”bubblar” till elementets förälder. Därefter skall filen ajax/savecomment.php anropas med AJAX. Datatypen skall vara JSON och det är med POST metoden AJAX skall kommunicerar med servern på. Data som skall skickas till servern är id:et för den låt som kommentaren gäller samt kommentaren i klartext. Om något går fel skall detta meddelas användaren på valfritt sätt. Om allt går som tänkt skall den nya kommentaren visas som första kommentar under länken ”Show all comments”.
- När användaren klickar på länken ”Like [filnamn.ogg]” skall default-beteendet för länken avbrytas och likaså att klickhändelsen ”bubblar” till elementets förälder. Därefter skall filen ajax/savecomment.php anropas med AJAX. Datatypen skall vara JSON och det är med POST metoden AJAX skall kommunicerar med servern på. Data som skall skickas till servern är id:et för den låt som användaren gillar. Om något går fel skall detta meddelas användaren på valfritt sätt. Om allt går som tänkt skall siffran efter ”Count” uppdateras.
- När användaren klickar på bilden för en utsökt artist skall bilden vissas som fullstorlekt med SlimBox 2.