Projektrapport webbserverprogrammering 1

Av Samuel Gjekic

**Bakgrund**

Två delprov programmerade i PHP -en nedräknare och inloggnings sida- ligger som bakgrund till denna rapport.

**Problem**

I denna sida skulle jag behöva 2 tabeller i en databas, dessutom måste jag lära mig mer om JOIN och Prepared statements för att slutföra uppgiften på det sätt som jag vill göra.   
Jag måste kunna hämta data och ta bort data i tabellerna, men även redigera data.

För detta måste jag göra en klass som har alla dessa funktioner, på ett snyggt och effektivt sätt.

Sedan måste jag planera hur själva projektmappen ska se ut.

**Metod**

**Vattenfallmetoden, Agile, (Kanske en blandning av bägge två)**  
Jag utgick ifrån vattenfallsmetoden. Jag började med att konstruera upp själva webbplatsen på ett papper, jag kollade lite vad för typ av saker som behövdes osv. Sedan började jag med design, detta både projekt arkitekturen men även stilen som sidan använde sig av.   
  
Sedan började jag bygga på själva scripten, först klassen som skulle sköta all hantering av filmer. Jag integrerade detta på dom olika sidorna jag har. Testade mig fram, efter lite felsökning och testning osv så blev jag nöjd. Jag organiserade hela projektet och skrev denna rapport.

**Genomförande**

**Del 1 Problemlösnig**

Jag började med att göra en simpel config.php fil i mitt /includes/ folder. I denna fil använder jag mig av define för att definera vissa strings, så som page title m.m. Det gör det lättare om jag senare behöver ändra något, då behöver jag inte ändra det i alla filer utan endast i config.php. Detsamma gäller databasens credentials.  
  
Efter detta gjorde jag en simpel mall med hjälp av bootstrap och lite css kod, jag valde bootstrap då jag tycker det är smidigare för någon som mig, då jag inte gillar css för mycket.   
  
Index.php är själva huvudsidan, där har jag valt att ha både listan med filmer samt formuläret för att lägga till nya filmer, i själva tabellen över filmer har jag lagt till 2 knappar, en redigera knapp och en ta bort knapp, dessa använder jag mig av simpla parametrar i URL:n för att skicka ett GET request och sedan gå vidare efter de baserat på vilken knapp användaren har tryckt.

Men innan jag började med allt annat, så skapade jag en Movies\_Module klass, movies\_module.php. Denna klass innehåller alla funktioner som har med filmer att göra, t.ex spara till databas, ändra, ta bort osv... Detta gjorde jag för att ha mer koll på koden.   
  
Om användaren trycker Redigera i index.php kommer man till edit.php, dock med en parameter, id. Detta gör att jag lätt kan store denna ID på edit.php sidan, och jag kan även använda den för att hämta filmen och fylla i edit formuläret så att det ger en snyggare upplevelse för användaren. När användaren trycker spara ändringar så används ***updateMovieById*** metoden i movies klassen, denna funktion tar några parametrar och sedan använder prepared statement för att uppdatera filmen i databasen, sedan blir man skickad till index.php. I början av scriptet körs även ***getMovieById*** för att hämta den nurvarande filmens information.   
  
Om användaren trycker ta bort film så används en GET parameter som jag valt att döpa till action. Samt en ID parameter, sedan tas filmen bort från databasen, med hjälp av ID i URL vet vi vilken film som ska tas bort, tillslut skickas användaren tillbaka till index.php, vilket får själva processen att se lite bättre ut då även tabellen med filmer uppdateras.   
  
Lägger användaren till en ny film, använder vi oss även där av POST metoden, vi kollar först om någon Post metod har skickats, vi tar emot alla värden och använder oss av htmlspecialchars för att minska säkerhetsrisker, dock vet jag att man inte behöver göra detta kanske när man använder sig av prepared sql statements, då det konverteras till htmlspecialchars då ändå. När vi samlat all data från POST metoden som vi behöver, skickar vi det till ***addMovie*** funktionen i movies klassen, där med hjälp av ett prepared statement så lägger vi till en ny film i databasen. Sedan skickar vi användaren till index.php igen så att sidan laddas om, för en bättre användarupplevelse.  
  
För att visa alla filmer från databasen, använder jag mig av kanske inte den bästa lösningen, men jag har en funktion i movies klassen, ***getAllMovies*** , denna använder sig av join i ett prepared statement för att binda själva kategori id till ett annat table i databasen, som heter categories. Mer om det nedan. Men efter den hämtat alla skickar vi tillbaka en array som vi lagrat alla filmer på, vi har även felhantering och gör att om det inte finns några filmer skickas det tillbaka en tom array.   
  
Själva databasen är simpel, det är en databas, 2 tables, ***movies*** och ***categories*** . När vi hämtar data från movies, binder vi det ID som är i "category" rutan, till den kategori som står skriven med samma id i ***categories*** tabellen.

**Del 2. Utvecklings möjligheter**

Det finns vissa säkerhetsalternativ såklart som kan anpassas, sen är det ju fritt fram för vem som helst att t.ex ta bort eller lägga till filmer på sidan om den är live, att ha någon slags authentication på sidan som bestämmer vad man får göra och med vad vore ju bra, detta kan t.ex lagras i session data eller cookies.   
  
Stilen kan ju förbättras, själva koden kan nog bli stiligare men jag kände att jag var nöjd så. Dessutom är ju kommentarerna skrivna på Svenska, jag tycker det är bäst att skriva dom på Engelska såklart men jag ville få fram mina kunskaper till er så valde att använda mig av Svenska kommentarer denna gången.   
  
Man skulle även kunna dela upp själva Movies\_Modules.php klassen, kanske genom att extenda den, detta skulle resultera i mindre kod i varje dokument, för just nu är det runt 100 rader kod.

**Del 3. Reflektioner**

Jag känner att jag gjort en bra insats, jag tror jag fått med det som behövdes för ett högt betyg. Mina planer i början var dock att använda mig av javascript för att få allt att uppdateras utan att uppdatera själva sidan, men jag kände att detta inte var passande för själva uppgiften som ska vara i PHP. Har både använt mig av JOIN och Prepared statements, jag har även gjort endel felhanteringar som gör att det inte borde bli några errors på själva sidan om något går fel. Jag är nöjd med min insats.

**Bilagor**  
  
Funktionen som hämtar enstaka film med hjälp av id:  


Funktionen som hämtar alla filmer från databasen:



En funktion som tar bort film och en som lägger till film i databasen:



Funktion som uppdaterar en existerande film i databasen:

