Rapport – film bibliotek

Webbserverprogrammering 1 – Izabela Marcinkowska

Innehållsförteckning

[Förberedelse 2](#_Toc138690718)

[Genomförande 2](#_Toc138690719)

[Skapa databas 2](#_Toc138690720)

[Index 3](#_Toc138690721)

[Add movie 3](#_Toc138690722)

[Ändra film 4](#_Toc138690723)

[Edit movie 4](#_Toc138690724)

[Uppdate movie 4](#_Toc138690725)

[Radera film 5](#_Toc138690726)

[Delete film 5](#_Toc138690727)

[Confirm delete 5](#_Toc138690728)

[Utvecklingsmöjligheter 5](#_Toc138690729)

[Systemutveckling 5](#_Toc138690730)

[Reflektioner 6](#_Toc138690731)

[Utmaningar 6](#_Toc138690732)

# Förberedelse

Jag började med att skapa en plan, att göra lista för att kunna genomföra projektet. Jag redan från början visste hur jag vill att programmet ser ur och vad som behövs göras. Jag har skapat enkel att göra lista för att skapa struktur i mitt arbete och kom fram till enkel översikt:

En bild som visar text, skärmbild, Teckensnitt, programvara

Automatiskt genererad beskrivning

Jag behövde träna lite mer på prepared statements som jag inte har infört i mitt förra projekt och det var första jag fokuserade på för att kunna få högre betyg. Jag har snabbt förstod syfte med dessa och ville implementera dessa i mitt projekt för att öka säkerheten i mitt program.

Efter det satte jag i gång med att utföra uppgifter.

# Genomförande

## Skapa databas

Jag började med att skapa databas i phpMyAdmin på min dator. Sedan behövde jag skapa tabeller i databasen som kommer att hålla filminformationer. Detta script används endast i början för att skapa uppgifter i databasen och sedan behövs den inte mer. Jag gav filen namn generate.php och dess funktion är endast för att skapa enkel databas med olika exemplar på filmer som man kan jobba på.

Jag skapade två tabeller som sedan kopplade jag ihop med id nummer med FOREIGN KEY, d.v.s. att id nummer av genre i tabellen ”movies” är direkt kopplat med id nummer i tabellen ”genres”. I tabellen ”movies” sparas informationen om genre i form av id nummer - int. Sedan i tabellen ”genres” representeras id en specifik genre i form av en string. Detta gör jag till exempel på grund av att om man skulle vilja addera eller editera namnet på genre så behöver man inte justera varenda film som har denna genre utan man endast behöver ändra detta i tabellen genre och alla filmer med denna specifika id kommer att ändras.

Sedan körde jag dessa queries med if satser att kontrollera att dessa skapas korrekt. Vid fel ska information visas för användaren.

För att man skulle kunna arbeta med något så skapade jag ”samples” med olika filmer som per automatik sparas i databasen. För detta använde jag en array som sedan med hjälp av foreach funktion gå igenom varje element i array som sparar den i databasen. Där var det lite klurigt just med genre. Eftersom jag använde id nummer i ”movies” tabellen så kunde jag inte bara skriva ”Komedii” i array för genre. Där med hjälp av en ytterligare query behövde jag ta fram från kolumnen ”genres” id nummer som representerar angiven genre i array. Sedan efter det med ytterligare query skickas alla uppgifter, tillsammans med korrekt, i form av int till databasen.

## Index

Jag har bestämt mig att för index sida inte fokusera så mycket på design och css utan använde mig av redan existerande bibliotek med enkel css design och jag har kopplat detta i index filen. Jag använde Pure CSS bibliotek. (<https://purecss.io>)

Jag skapade en enkel form som man kan fylla med title, director och year. Sedan efter det behöver man välja genre av filmen. Där fanns flera olika steg jag behövde göra att få detta fungera på det sätet jag ville. Jag ville att listan skulle ändras varje gång man ändrade tabellen ”genres” med nya genres. Detta behövde jag först ta fram alla existerande genres från genre kolumnen först och sedan ange dessa i html radio-input.

Under formen ville jag ha listan med filmer som redan finns i databasen, för denna stund endast sample filmer skapade i början. Detta gjorde jag med hjälp av en tabell och uppgifter i databasen tog jag fram med hjälp av while loop som gått igenom varje element i tabellen och skrivit ut dessa i tabellen i index. I tabellen har jag också angivit länkar till att ändra på filmen och radera som sedan har jag skrivit separata scripts för.

## Add movie

Nästa steg var att skapa script som adderar nya filmer i databasen. Detta script sätts i gång när användaren trycker på knappen ”Add movie” i index sidan. När detta händer så med hjälp av POST variabel tar vi fram det var som har skrivits i form fönster och sparar i variabler.

För säkerheten har jag använd många metoder att förhinda sql injections. Jag använde både sanitize funktion som jag själv skapade och sparade i db.php filen. Där går input genom flera olika ”rensning” funktioner som att ta bort slashes och ändra html taggar till bokstäver. Efter detta sparas dessa och går igenom ”prepared statements” del där anger vi vilken sort av input förväntar vi oss från användaren och till slut spara dessa i placeholders som vi har angivit. Till slut skickas dessa som query till databasen. Allt detta för att förhindra några sql injections och felaktig användning.

## Ändra film

### Edit movie

Nästa steg var att bygga funktion att ändra på filmen i databasen. Först skapade jag filen edit\_movie.php som laddas upp när användaren trycker på edit bredvid filmen. För sidan att veta vilken film som ska ändras har jag använt GET variabel som tar fram id nummer för filmen från url adressen. Då vet vi att det är just denna film som ska databasen ta fram uppgifter om. Om nu skulle inte finnas något id, eller den skulle vara felaktig så kommer användaren vidareskickats till index eller informerats om att filmen inte finns i databasen.

Med query tar vi då fram vilken film som ska tas fram och sedan dessa information visas på sidan i en form som nu användaren kan ändra.

Med generes behöver vi göra samma som i index att vi går igenom varje genre som finns sparat i genre tabellen och sedan skriver ut varje en med while funktion.

Efter det tar uppdate\_movie.php över.

### Uppdate movie

Med information som sparades i edit movie formulär använder jag återigen POST variabel att ta fram informationen som skickades. Dessa precis som i Add movie scriptet får gå igenom santirizing och prepared statements för att förhindra användaren från att kunna påverka databasen på otillåtet sätt.

Efter det används en annan query – UPPDATE som i stället för att addera, hittar den filen med angiven id och sedan ändrar values. Allt såklart under kontroll av prepared statements som ser till att korrekt sort av data har blivit angiven.

## Radera film

### Delete film

Här gjorde jag nästan likadant som med edit\_movie.php där jag först tog fram id från url och med hjälp av denna id kan programmet nu ta fram uppgifter om filmen från databasen och skriva ut dessa.

Under utskrivna information om filmen finns nu knappen som då raderar filmen från databasen. För detta behövde jag skapa en form med två inputs. En hidden och en som en knapp. Detta används för att genom hidden input kunna skicka informationen vidare till confirm\_delete.php om vilken film ska nu raderas. Användaren däremot ser endast knappen med bekräftelse.

### Confirm delete

Här finns det script som raderar filmen från databasen. Jag använder här också prepared statements och sanitize funktioner för att säkerställa att ingen har ändrat values under input från förra sidan. Säker query sedan skickas till databasen och med hjälp av DELETE raderas den från databasen.

# Utvecklingsmöjligheter

För att kunna förbättra programmet skulle man kunna först och främst addera knappen avbryt till att radera och ändra sidorna. Detta tyckte jag att jag inte längre har tid för eftersom tiden började att rinna ur. Man skulle kunna såklart göra mer av CSS och göra designen lite finare än den är just nu men jag tyckte inte att detta var prio just nu och använde mig snabbt av redan existerande bibliotek. Just nu är det också möjligt att spara flera likadana titlar. Man kanske skulle kunna förhindra detta från att hända med ytterligare tabeller. Man skulle kunna också skapa möjlighet att markera filmen som att man har sett den eller vill se i framtiden. Detta skulle vara bra lyft för programmet.

# Systemutveckling

Jag har under projektets gång använt mig mest av Vattenfallsmetoden. Detta på sätt och vis på grund av att vi inte kan skicka endast del av programmet till er. Det är metoden som jag är mest van vid och just i detta projekt som jag själv utvecklar så tycker jag att den är passande.

Eftersom mitt projekt är relativt litet där jag endast behövde tre till fyra funktioner som att skapa databas, skriva ut den, ändra på den och radera. Om jag skulle använda andra metoder som till exempel agil så skulle jag tekniskt sätt först behöva släppa ut databasen som endast visar ut filmer, sedan ge funktionen till att ändra och sedan till slut metoden som gör möjligt att radera filmen. Eftersom projektet är ganska liten så skulle det kanske inte vara optimalt men vid större projekt där det behövs månader för att kunna implementera sida eller program så är det definitivt bättre sätt.

Vid vattenfallsmetoden skulle vi behöva skriva hela programmet först nästan utan att kontrollera detta och sedan till slut när alla delar av programmet är klara då börja att jobba med att testa programmet. Detta skulle kunna orsaka massor med extra tid som behövs att hitta nu tillbaka till fel som vi gjorde och hitta vad inte fungerar.

Jag har såklart under arbetets gång inte definitivt arbetar med vattenfallsmetoden eftersom när första delen skapades har jag kontrollerat om det fungerar med att visa alla filmer. Sedan efter att jag implementerade edit funktion har jag kontrollerat om detta fungerar också, osv. Det är på sätt och viss också agil arbete men mer ”bakom kulisserna” eftersom inget av dessa små funktioner har blivit ”släppta”.

# Reflektioner

Jag tycker att arbetet med projektet gick bra, även om den hamnade under Midsommarfirande ☺

Flera delar av programmet har vi lite gjort förut i övningar men här behövdes det att ta alla samtidigt och logiskt tänka fram på vad som behövs göras och hur alla script, sidor och delar ska fungera med varandra.

## Utmaningar

Säkert större utmaningar som vi endast tog i böcker som behövdes skrivas nu var att koppla två olika tabeller med varandra med FOREIGN KEY och sedan anropa dessa två tabeller varje gång vi behövde ta fram, ändra eller radera information. Där med hjälp av JOIN behövde jag ta fram namnet på genre från en annan tabell. Detta tog såklart en hel del tänkande, googlande och fel.

Sedan var prepared statements också en liten nyhet eftersom dessa använde jag inte i förra övning och behövde därför läsa först hur logiken gick bakom detta. Detta tog kanske inte lång stund att förstå men krävde en hel del annan tänkande när man behöver använda dessa.

Allmänt tycker jag att det var en av största projekt jag hittills gjorde i NTI skolan direkt med den stora HTML och CSS projektet jag har gjort i Webbprogrammering. Jag tyckte också att PHP var riktigt kul språk att koda, den är enkel och förlåtande till skillnad till exempel från C# som jag hade i början av året.