

Christoffer Goude - 20140607

Individuellt mjukvaruutvecklingsprojekt - Slutrapport

Fully Awesome

[Christoffer Goude - 20140607](#)

[Individuellt mjukvaruutvecklingsprojekt - Slutrapport](#)

[Fully Awesome](#)

[Abstrakt](#)

[Inledning](#)

[Positiva erfarenheter](#)

[Negativa erfarenheter](#)

[Sammanfattning](#)

Abstrakt

I denna rapport kommer jag gå igenom det projekt jag utfört under dom senaste sju veckorna. Fokuspunkterna för rapporten kommer vara både dom nya kunskaper jag har tagit till mig under projektets gång, samt dom negativa erfarenheterna som jag stött på.

Den största slutsatsen jag har dragit av projektets resultat är att en risklista kan vara otroligt viktigt för att kunna ro projektet i hamn inom deadline. Genom att skapa en risklista tidigt kan man utröna vilka delar av projektet som kommer vara problematiska. Detta kan göra att man från början av projektet kommer kunna lägga ett huvudsakligt fokus på dessa delar och bättre kunna fördela sin tid.

Inledning

Bakgrund

Den produkt jag skapat under projektiden är en webbdesignblogg kallad Fully Awesome. Bakgrunden till detta val är att jag vill kunna skriva om många av dom intressanta tips och tricks som rör webbdesign som jag dagligen läser om online. Det finns många "färdigbyggda" alternativ för att kunna blogga i dagsläget, men som webbprogrammerings-student tycker jag det var ett självklart val att skapa en egen blogg. Detta skulle ge mig en god erfarenhet, en möjlighet att utöka mina kunskaper, och även full kontroll över alla förändringar på sidan i framtiden.

Målet med projektet var att skapa en sida där besökare kan läsa de inlägg som jag (administratören) lägger upp. Besökarna ska även ha möjlighet att genom en enkel registrering skapa en användare och då få möjlighet att kommentera på inläggen. Detta valet gjorde jag då den målgrupp jag har inriktat mig på är webbdesigners och programmerare som just vill läsa om dessa områden och även komma med egen input. Genom att kunna lägga en kommentar på inläggen får de möjlighet att föra en diskussion med andra som har samma intresse.

Denna blogg har jag tidigare skapat i kursen Individuellt Mjukvaruutvecklingsprojekt förra året, men blev inte färdig och heller inte nöjd med resultatet. Därför valde jag att göra om samma projekt i år, men har nu valt att byta till andra programmeringsspråk som jag känner mig mer bekväm med så jag kunde tillföra mer funktionalitet till sidan.

Teknik

Först och främst har jag använt HTML för grunden av sidan och CSS för designen. Med hjälp av PHP hanterar jag databasanslutningar och sessionshantering för inloggningen av användare. För att skicka och hämta data mellan sidan och datahanteringslagret använder jag Javascript, jQuery och Ajax. Till sist använder jag ett ramverk som heter Skeleton. Detta ramverk var nytt för mig, men visade sig vara både lätt att använda och ändå relativt kraftfullt.

Positiva erfarenheter

I detta projekt har jag lagt mer fokus på responsiv design och anpassning för olika surfplattformar än vad jag har gjort tidigare. Detta har gett mig mycket bra erfarenhet, och är något som jag känner är väldigt viktigt att tänka på för dagens webbutvecklare. Skeleton är ett mycket bra ramverk som lägger en otroligt bra grund för responsiv design av webbsidor, och jag kommer definitivt använda det i fler projekt.

Negativa erfarenheter

Under projektets gång har dom största problemen jag stött på handlat om anslutningen till min databas. Jag använde ett för mig nytt sätt att ansluta till databasen, nämligen Php's PDO, och trots att det såg enkelt ut på papperet så blev det en hel del problem med detta.

Först och främst hade jag bestämt mig för att använda ett ramverk kallat Medoo för att sköta databasanslutningarna, men då detta inte fungerade alls så enkelt som det verkade kunna göra slopade jag det helt och hållet. Efter lite efterforskning bestämde jag mig sen för att använda Php's inbyggda funktion PDO (PHP Data Objects). På papperet verkade PDO vara mer kraftfullt än till exempel PHP's andra inbyggda interface för databasanslutningar, MySQLi, då PDO till exempel stödjer fler databastyper.

Efter att ha studerat hur anslutningen via PDO skulle gå till implementerade jag den första PDO-koden, men fick inte anslutningen att fungera. Genom att fånga upp felmeddelanden kunde jag felsöka min anslutning och implementera lösningar för dessa problem, men detta tog upp mycket av min tid. Då det uppstog flera olika problem slutade det med att jag hade lagt långt mycket mer tid på att bara skapa en stabil anslutning till databasen än vad jag hade väntat.

Inför nästa större projekt jag ska göra planerar jag att skapa en tydlig risklista för vilka delar av projektet som skulle kunna ta lång tid och på så sätt kunna avsätta mer tid till dessa delar redan tidigt i projektet. Om jag hade gjort en genomtänkt risklista denna gång hade jag möjligtvis kunnat snappa upp att databasanslutningen skulle kunna bli problematisk och lagt mer tid på efterforskning gällande detta problem.

Sammanfattning

Under detta projekts gång har jag stött på en del motgångar och förseningar, men överlag känner jag att jag fått ut mycket kunskap från dessa motgångar. Jag har också tagit mig an vissa delar av webbdesign som jag tidigare bara "skrapat på ytan".

I framtiden vill jag fortsätta utveckla min sida, och självklart börja använda den så som det var tänkt att den skulle användas. Jag har redan börjat planera vad jag ska utöka sidan med, först och främst vill jag skapa ett mer robust administrationssystem. I dagsläget kan administratören endast lägga till inlägg på sidan, men behöver man ta bort inlägg eller administrera användarna på sidan behöver detta ska direkt i databasen. Att lägga till funktionalitet för detta står högt på min lista.

Utöver detta vill jag också se över strukturen för koden på sidan. För närvarande ligger mycket av koden i endast två filer, och detta skulle kunna ändras på genom att dela upp koden i stycken

och lägga varje stycke i en separat fil som hanterar en separat funktionalitet på sidan. Detta skulle ge en bättre överblick av koden och ett sånt tillvägagångssätt skulle göra det enklare att både lägga till och ta bort funktionalitet i framtiden. Om man till exempel av någon anledning vill ta bort möjligheten att kommentera inlägg kan man ta bort dom filer som hanterar kommentarerna, och på samma sätt lägga till nya filer som hanterar ny funktionalitet, och som i sin tur ansluter till databasen via ett separat databas-lager.

Utöver detta känner jag att det finns mycket potential för att lägga till mer och mer funktionalitet på sidan. Detta är ett projekt jag kommer kunna ha glädje av långt framöver, och en sida som jag kommer kunna använda för att finslipa mina webbprogrammeringskunskaper med.