Rapport – Webshop med backend

Eksamensprojekt i Webprogrammering Server Side

Udarbejdet af Stine B. Holt og Marie B. S. Therkelsen

Hold: cs16auwssea3 – CPHbusniess Academy

Afleveret d. 23. December 2016 på [www.wiseflow.dk](http://www.wiseflow.dk)



Indholdsfortegnelse

Projektbeskrivelse 3

Sidens funktioner 4

Arkitektur 4

Databasen 8

Sikkerhed og performance 9

Konklusion 9

# Projektbeskrivelse

Vi har fået til opgave at programmerer en webshop i NetBeans, med backend i form af en database der er programmeret i MySQL.

Kravene til siden er at den skal kunne følgende funktioner:

* Vise de tilgængelige varer hvor der er flere detaljer omkring produktet
* Login
* Registrering af nye brugere med for/efternavn, e-mail, mobil og adresse
* Kurv side som vise hvilke varer, der er lagt i kurven
* Check ud funktionalitet hvor folk kan ”betale”, dog uden den sidste side med kreditoplysninger

Tekniske krav:

* Data skal persisteres i en database
* Der skal bruges MVC (model-view-controller design) lagdeling, hvor:
  + Præsentationslaget laves med JSP
  + Kontrollaget laves med Serlets
  + Modellaget laves med javaklasser
* JSP siderne kan gøre brug af expression language og JSTL core biblioteket eller blot bruge scriptlets
* Login skal gemmes i en session således at brugeren er logget ind på tværs af sider

Indholdet af rapporten har vi valgt at holde forholdsvis komprimeret, da det giver et bedre overblik over vores webshop og vi skal begrænse rapporten til 14.000 tegn, der henvises derfor til kildekoden, hvis der er ønske om at se koden for den enkelte metode. Dog kan der forekomme enkelte kodeeksempler i rapporten.

Vores webshop sælger plakater i forskellige kategorier, med levering. Her kan kunderne altså købe en flot plakat til hjemmet og få den leveret lige til døren.

Som kunde har man sin egen personlige konto på webshoppen, derfor kræver det et login, med brugernavn og password for at kunne handle på sitet.

## Sidens funktioner

Når man ønsker at tilgå webshoppen bliver man mødt af en login side, her logger man enten ind med sit brugernavn og password eller man klikker på knappen ”Sign in” som videresender dig til en side hvor det er muligt at oprette sin egen personlige bruger på webshoppen, når alle oplysninger er udfyldt får man adgang til webshoppen.

Her bliver man mødt af en pæn og overskuelig siden, hvor der er fremvist billeder af plakater, navnet på plakaten, kategori og pris på plakaten, hvis man har et ønske om at få flere detaljer om plakaten kan man klikke på knappen ”Se mere” som findes under hvert produkt. Her sendes man til en side som indeholder yderligere information om plakaten. Her kan man tilføje sit produkt til indkøbskurven eller gå tilbage til oversigtssiden med alle plakaterne.

Hver gang man tilføjer et produkt til sin indkøbskurv, vises der en advisering der informere om at der nu er tilføjet et produkt til kurven. Når man så har tilføjet alle de plakater man ønsker at købe klikkes der på knappen ”Indkøbskurv”, her vises alle de plakater man har tilføjet til kurven, med navn, pris og ikke mindst summen af alle plakater i indkøbskurven. Herefter kan man så klikke på ”bestil ordre” også sendes man tilbage til webshoppens oversigtsside og der er en advisering i højre hjørne om at ordren nu er bestilt.

Udover disse funktioner viser webshoppen også hvem man er loggen ind som og har en logout funktion.

## Arkitektur

Vores website er opdelt efter MVC-modellen som er et program der er baseret på hændelser, hvilket vil sige at en indtastning i brugergrænsefalden f.eks. resulterer i en hændelse til controlleren, altså foregår kommunikationen mellem datamodellen og brugergrænsefalden via en controller der adskiller de to dele fra hinanden, hvor det derfor også er lettere at vedligeholde koden, da tingene er opdelt og overskueligt at gennemskue i koden.

Opdelingen af MVC:

* Modellen er f.eks. en database eller andre datakilder
* Controlleren håndterer logikken
* View’et indeholder de visuelle elementer(brugergrænsefladen)

Purchasekurv.jsp

ProduktDetails.jsp

produktside.jsp

createNewUser.jsp

Login.jsp

VIEW-laget

Control-laget

BuyProduct.java

showProducts.java

ProductDeail.java

loginServlet.java

createNewUserServlet.java

showPurchaseServlet.java

placeOrder.java

View-laget indeholder brugergrænsefladerne for de forskellige views på jps-sider:

* **Login.jsp** – Brugeren logger ind og bliver sendt videre til login servletten (login.java) og link til oprettelse af en ny bruger via createUser.jsp
* **CreateUser.jsp** –Brugergrænsefladen til oprettelse af en ny bruger hvori parametrene fornavn, efternavn, e-mail, mobil og adresse indtastes.
* **Produkt.jsp** – Brugergrænsefladen for visning af alle produkter og muligheden for at tilføje dem til indkøbskurven, når et produkt er valgt benyttes BuyProduktServlet
* **PurchaseKurv.jsp** – viser listen med de valgte varer, pris og antal. Summerer prisen på alle valgte vare. Action submit ordre (placeorder), sender derefter en besked til brugeren om at ordren er gået igennem.

Control-laget indeholder vores servlets, som er et javaprogram, der modtager forespørgsler over internettet. Her finder man:

* **Login.java** – Authenticate brugeren via UserMapper (authentcateUser),

IF authenticated

Hentes listen med produkterne fra getAllProdukts  
Else

Sendes en besked til brugeren om at de ikke er logget ind og der er mulighed for at oprette en ny bruger via newUser

* **NewUser.java** - Benytter UserMapper til at oprette nye brugere (createUser), og herefter sender brugeren til login.jsp efter oprettelse
* **BuyProduktServlet.java** – Placerer varerne i indkøbskurven og trækker parametrene fra requestet(productID, UserID)
* **ShowPurchaseServlet.java** – Generere en liste for brugerens køb (getAllPurchaseByUser), når submit action fra indkøbskurven bliver sendt placeres ordren (placeorder)

Model-laget indeholder vores mappere og dataaccess funktioner, her finder man:

* **DB.java** – Som er vores database connection til webshop databasen.
* **UserMapper** – Indeholder flere vores metoder omkring brugere og login;
  + authenticateUser – Funktion til at logge ind med eksisterende bruger.
  + createUser – Opretter nye brugere
  + getUser – Vælger den ind loggede bruger i forbindelse med køb af en vare.
* **ProduktMapper** – Indeholder vores metoder i forbindelse med visning af produkter/varer og detaljer omkring disse.
  + getAllProducts – Indeholder listen med alle vores produkter
  + getProduct – Indeholder en detaljeret side om alle produkterne
* **PurchaseMapper** – Indeholder metoder omkring indkøbskurven
  + placeProduktForPurchase– Placerer valgte varer i indkøbskurven
  + getAllPurchaseByUser – Viser indkøbskurven/listen med alle valgte varer, herunder produkt Id, produkt navn, antal og prisen
  + placeOrder – Submitter alle varer i indkøbskurven, (her ville man i andre tilfælde, blive sendt videre til en betalingsside, men den del skal vi ikke lave, derfor godkendes der bare at det er de varer brugeren vil have).

## Databasen

Vores database er udarbejdet i MySQL og opdelingen er:

I vores database har vi 5 tabeller;

* **Image –** Indeholder billederne der vises i brugergrænsefladen ud fra hver vare;
  + Image via blob
  + Name
  + extenstion
* **Produkt –** Indeholder alle vores produkter(plakater) og detaljerne omkring disse;
  + Id
  + Name
  + Kategori
  + Description
  + Size
  + Pris
  + Leveringstid
  + Image
* **User –** Indeholder alle brugerprofiler med oplysninger om brugerne;
  + ID
  + Username
  + Password
  + FName
  + LName
  + Email
  + Mobile
  + Adresse
  + Postnummer
* **Ordertable –** Indeholder alle ordre der er lavet på webshoppen af brugerne;
  + OrdreID
  + Username
  + Ordredate
  + Ordrebestilt
* **produktONorder –** Indeholder de produkter der er lavet en odre på
  + ProduktID
  + OrdreID

Derudover har vi lavet nogle SQL-sætninger som ”viser data pr. ordre” og ”summen af ordrens pris for en bruger”, som vises på PurchaseKurv-siden.

# Sikkerhed og performance

I forhold til sikkerhed har vi hashet vores password i databasen, hvilke vil sige at man ikke kan læse passwords i databasen men de vises som en række tal og bokstaver. Ved at bruge hashing undgår man at hackere vil kunne skaffe sig adgang til vores database hvor alle webshoppens brugeres brugernavne og passwords ligger gemt i klartekst, ved at hashe krypterer man passwords så de ligger i talkoder der ikke kan bruges som passwords.

Derudover kunne man sikre webshoppen for man-in-the-middel angreb, som ville kunne gøres ved at sætte siden op med HTTPS som er en kommunikation over HTTP, som er en protokol der bruges til kommunikation på World Wide Web (WWW), dog uden en krypteret forbindelse hvorimod HTTPS er en krypteret forbindelse der beskytter mod et angreb hvor en tredjeperson udgiver sig for at være afsender/modtager. Dermed kan tredjepersonen ændre på kommunikationen mellem afsender og modtager, f.eks. en bruger og en server, der tror de kommunikerer med hinanden og ved at tredjepersonen udgiver sig som en af parterne lokkes der ofte personfølsomme oplysninger ud af en af parterne der vil blive misbrugt af tredjepersonen.

# Konklusion

Vores webshop fungerer som den skal, koden er letlæselig og ordentlig i strukturen. Vi har opfylde alle de obligatoriske krav der var stillet til projektet og rapporten.