

Kunskapskontroll 3: E-butik, gruppuppgift

Läs uppgiften noggrant innan ni sätter igång!

Uppgiften handlar om att skapa en E-butik där databaskoppling och databashantering görs via PDO med MySQL. Applikation skall vara uppdelat i en publik sektion och ett användargränssnitt för administratör, precis som föregående blogg-uppgift.

Uppgiften är en gruppuppgift om 3 personer, varken mer eller mindre. Ni väljer era partners och rapporterar detta till mig. Självklart skall projektet versionshanteras, ni väljer vems Github konto som skall hantera projektet.

Ni utvecklar projektet utifrån en befintlig mall "ecommerce-template.zip", som sätter grunden för hur mappstrukturen skall se ut i ert projekt. Detta för att hjälpa er komma igång och för att behålla en god kodstruktur genom hela utvecklingsprocessen.

E-butik är ett alldeles för stort system för att kunna utvecklas under detta projekt. Därför är endast vissa utvalda delar inkluderade och obligatoriska i denna uppgift.

Obligatoriska delar

Produkter

- Shop:
 - Visa lista på produkter (READ)
 - Visa produktsida (READ)
- Admin
 - Visa lista på produkter (READ)
 - Befintlig produkt skall kunna uppdateras (UPDATE)
 - Befintlig produkt skall kunna raderas (DELETE)
 - Nya produkter skall kunna skapas (CREATE)

Användare:

- Shop:
 - Användare skall kunna registrera ett användarkonto (CREATE)
 - Användare skall kunna logga in på deras användarkonto (READ)
 - Användare skall kunna navigera till “mina sidor”, en lösenordsskyddad sida (READ)
 - Användare skall kunna se sina uppgifter i “mina sidor” (READ)
 - Användare skall kunna uppdatera sina uppgifter i “mina sidor” (UPDATE)
 - Användare skall kunna radera sitt användarkonto i “mina sidor” (DELETE)
- Admin
 - Visa lista på användare (READ)
 - Befintlig användare skall kunna uppdateras (UPDATE)
 - Befintlig användare skall kunna raderas (DELETE)
 - Nya användare skall kunna skapas (CREATE)

Köp processen (Denna går vi igenom tillsammans. Ni implementerar denna del genom att följa videogenomgångarna)

- En fungerande varukorg, där produkter kan läggas till
- Valda produkter kan hanteras på kassasidan
- Order placering, där information om produkter, användare och annan väsentlig information sparas i DB tabellerna orders och order_details

Frontend/Design

- En enhetlig design på både shop-del och admin-panel
- Huvudmeny på både shop-del och admin-panel
- Start sida för shop-delen
- Varukorg (frontend-del)
- En kassa sida med lista av produkter och formulär för användaruppgifter (frontend-del)

Databasens obligatoriska tabeller och kolumner

De obligatoriska tabellerna och kolumnerna ser ni nedan. Ni får absolut inte ändra på de obligatoriska kolumnerna såsom inställningar och namngivning. Ni får absolut inte bortse från någon av kolumnerna. Notera dock att ni får lägga till extra kolumner och tabeller, för frivillig utbyggnad, utöver de obligatoriska delarna.

Notera att "img_url", i products-tabellen är den enda kolumnen som är NULL. NULL är en inställning som tillåter kolumnen att vara tom. Dvs inget krav med att implementera funktion för produktbild. Observera att resterande kolumner i alla tabeller är obligatoriska. Dvs resterande kolumner måste få ett värde för att skapa nya Produkter/Användare/Ordrar

products

- id, INT (9), Primary Key, Auto Increment
- title, VARCHAR (90)
- description, TEXT
- price, INT (9)
- img_url, VARCHAR (255), NULL

users

- id, INT (9), Primary Key, Auto Increment
- first_name, VARCHAR (60)
- last_name, VARCHAR (60)
- email, VARCHAR (150)
- password, VARCHAR (60)
- phone, VARCHAR (60)
- street, VARCHAR (255)
- postal_code, VARCHAR (255)
- city, VARCHAR (90)
- country, VARCHAR (90)
- register_date, TIMESTAMP, CURRENT_TIMESTAMP

Tabellerna orders och order_details kommer jag skapa i samband med genomgången av kassa/varukorg/orderplacering.

Ni skapar dessa tabeller genom att följa videogenomgångarna.

Tidtabellen ser ut som följande

20/5- 29/5

Användare och **Produkter** skall fungera både i Shop-delen och adminpanelen. Fokus är på att det fungerar. Styling får ni tänka på i efterhand.

Mycket av det ni gjorde i bloggen kan ni ta med er till denna uppgift, med viss justering.

Särskilt hur ni hanterade CRUD och databaskopplingen.

Även genomgången av my-page-2 ger er mycket bra information hur man arbetar med CRUD, användarregistrering och login. I my-page-3 förklaras även hur registrering och login fungerar med krypterade lösenord.

1/6- 5/6

Varukorg, produkthanteringen i kassa Jag utvecklar dessa, som ni kommer kunna implementera via videogenomgångarna. Notera ni får inte koden, ni får endast ta del av koden via videogenomgångarna, som spelas in.

- En fungerande varukorg, där produkter kan läggas till
- Valda produkter kan hanteras på kassasidan
- Order placering, där information om produkter, användare och annan väsentlig information sparas i DB tabellerna orders och order_details

8/6- 12/6

Färdigställ frontend delen både i admin och shop-delen. Många av dessa delar har ni redan utvecklat i föregående kurs. Ta vara på de redan färdigutvecklade delarna, särskilt vid tidsbrist.

15/6- 18/6

- Justera och färdigställ arbetet.
- Förbered och träna inför redovisningen.
- Gruppredovisning på torsdag den 18/6

Betyg

G

Bedömning sker efter studentens kod i Github, samt redovisningen.

- Studenten implementerar CRUD (CREATE, READ, UPDATE, DELETE) för någon av e-butikens obligatoriska delar (Produkter, Användare).
- Studenten jobbar aktivt med att färdigställa de obligatoriska delarna såsom frontend, köpprocessen (varukorg, kassa, order placering) osv

VG

- Studentens egna CRUD skall refaktoreras till antingen funktioner eller klasser
- Studenten visar på minst 2 AJAX anrop, i samband med sin egen CRUD hantering.

Deadline

Gruppredovisning online via teams, torsdag den 18/6.

Se till att ha med en SQL-exportfil i er github, så att jag kan importera er DB och testa er e-butik lokalt.