

Eksamen, DSDS, forår 2008

Introduktion til Scripting, Databaser og Systemarkitektur

Jonas Holbech
IT Universitetet i København*

3. juni 2008

- *Alle hjælpemidler er tilladte, dog ikke computer og kommunikationsmidler.*
- *Eksamen er skriftlig, fire timer (9-13), og den bedømmes fra 0 til 100 procent.*
- *Eksamen består af fem opgaver der alle ønskes løst.*
- *Skriv med kuglepen og kun på den ene side af papiret*
- *Eksamen må besvares på dansk eller engelsk*
- *Et godt råd: Gennemlæs opgavesættet inden du begynder at besvare de enkelte opgaver.*

Introduktion

I dette eksamenssæt skal du implementere dele af et fiktivt webbaseret system kaldet “TradeIT”.

TradeIT er en byttebørs i stil med Den Blå Avis hvor brugerne kan sætte ting til salg.

Personer der har oprettet sig som brugere kan sætte ting til salg og kommentere på de ting der findes i systemet.

Systemet har en login-funktion som du ikke skal implementere. Når brugeren er logget ind sættes en cookie kaldet `uid` der indeholder brugerens `userid` (se Figur 2). Ved at tjekke om `$_COOKIE['uid']` findes og indeholder et heltal kan du få brugerens `userid` og tjekke om brugeren er logget ind.

Servicen er kun for ansatte og studerende ved ITU.

*Adresse: Rued Langgaards Vej 7, 2300 København S, Danmark. Email: holbech@itu.dk.

Figur 1 viser et site-map for det ønskede system.



Figur 1: Site-map for det web-baserede system. Et screenshot (kasser) i diagrammet angiver sider, brugeren kan se i systemet. Hver sides navn er angivet ovenfor hvert billede. En ikke-annoteret pil i diagrammet angiver, at en bruger kan klikke på et link i en side for at springe til en anden side. En annoteret pil i diagrammet angiver, at brugeren ved at klikke på en knap forårsager, at php-scriptet annoteret på pilen bliver kørt på serveren, og at php-scriptet videresender brugeren til en ny side.

Baggrund:

Filen `index.php` er systemets forside og viser alle de ting der er sat til salg.

For hver ting (*item*) er prisen og antallet af kommentarer angivet. For hvert *item* er der et link til `show_item.php`. Linket sender et *item id* (*iid*) med til `show_item.php`.

`show_item.php` lister en længere beskrivelse af *item*'et samt alle dets kommentarer. Siden indeholder et link til `add_comment.php`. *iid* sendes med via linket.

`add_comment.php` indeholder en form til at indtaste kommentarer til et givent *item*. Scriptet sender indholdet af formen til `insert_comment.php`

Scriptet `insert_comment.php` modtager nye kommentarer, validerer brugerens input og sender brugeren tilbage til `show_item.php`

Bemærk: I det følgende antages det at funktionerne `error`, `mydb_connect`, `html_header`, `html_footer` og `check_heltal` er tilgængelige i filen `utilities.php`. Alle funktioner, du bliver bedt om at lave, er ligeledes tilgængelige i filen.

`mydb_connect`: tager ingen argumenter og opretter forbindelse til systemets database.

`error`: tager en streng som argument. Strengen udskrives og funktionen kalder `exit`, der stopper programmet.

`check_heltal`: tager et tal som argument og tjekker om det er et gyldigt heltal. Er det ikke kaldes funktionen `error` og programmet stopper.

Koden for `html_header` og `html_footer`:

```
function html_header($title){
    <!DOCTYPE ....
    <html>
        <head>
            <title>$title</title>
        </head>
        <body>";
}

function html_footer(){
    echo "</body></html>";
}
```

Opgave 1 (15 procent) - HTML

Opgave 1.1 (7 procent)

Din Opgave:

Opstil HTML-koden for selve formen, der vises af siden `signup.php` i Figur 1. Besvarelsen skal have formen

```
<form> ... </form>
```

Layoutet i formen er ikke relevant, tekstuelle beskrivelser er ligeledes ikke nødvendige.

Der er følgende krav til formen.

- Formens `action` attribut skal have værdien `create_user.php` og metoden `post`
- Formen indeholder seks formelementer
 - Et `text` felt med `name` attributten sat til `name`
 - Et `password` felt med `name` attributten sat til `pw`
 - Et `text` felt med `name` attributten sat til `email`
 - Et `text` felt med `name` attributten sat til `phone`
 - Et `select` felt med `name` attributten sat til `usertype`. Select elementet indeholder tre `option`-elementer:
 - * Første element har værdien `forsker`
 - * Andet element har værdien `underviser`
 - * Tredie element har værdien `studerende`
 - Et `submit` felt med `value` attributten sat til `Opret bruger`

Bemærk: På billedet i figur Figur 1 er `select` menuen foldet ud

Opgave 1.2 (8 procent)

Din Opgave:

Opstil HTML-koden for tabellen på siden `index.php` i Figur 1. Besvarelsen skal have formen:

```
<table ...> ... </table>
```

Teksten i første række skal fremhæves, enten med `<strong>` eller `<th>`.

Links i "Ting" kolonnen peger på `show_item.php` og sender et `iid` med i url'en.

- "Min gamle bamse" har `iid`: 1
- "Audi S8" har `iid`: 2
- "CSS for dummies (bog)" har `iid`: 3
- "2 biografbilletter" har `iid`: 4
- "SonyEricsson K800i uden batteri" har `iid`: 5

Opgave 2 (15 procent) - PHP

Opgave 2.1 (7 procent)

Din Opgave:

Skriv funktionen `chk_cookie`.

Funktionen tager ingen argumenter.

Funktionen skal tjekke om `$_COOKIE['uid']` er sat, og om den indeholder et gyldigt heltal.

Indeholder den et gyldigt heltal, skal tallet returneres. indeholder den ikke et gyldigt heltal skal du returnere `false`.

Hint: Husk når du benytter `return` afvikles resten af koden ikke.

Du kan få brug for funktionerne `ereg`, `isset` og konstruktionen `if`.

Opgave 2.2 (8 procent)

Baggrund:

`www.itu.dk` vil gerne have mulighed for at vise et tilfældigt *item*.

Din Opgave:

Skriv funktionen `random_item`.

Funktionen skal tage et *array* som argument og returnere et tilfældigt element fra *array*'et.
Hint: Husk funktionen *rand* tager to argumenter (min og max) og returnerer et tal mellem min og max, begge inklusive.

Opgave 3 (15 procent) - Regulære Udtryk

Opgave 3.1 (5 procent)

Skriv et regulært udtryk, der matcher danske telefonnumre.

Hint: Dvs. tal bestående af præcis 8 tegn, hvor første tal er større end nul.

Det regulære udtryk skal således matche strengene '35353193', '22123465' men ikke eksempelvis '02348264' eller '112'.

Opgave 3.2 (5 procent)

Baggrund:

Da servicen kun er for personer tilknyttet ITU, skal vi filtrere, hvem der kan tilmelde sig. Det kan vi gøre ved at tjekke om emailen er fra ITU.

Din Opgave:

Skriv et regulært udtryk, der matcher gyldige emails fra ITU.

Det regulære udtryk skal således matche strengene 'gitte_hansen@itu.dk', 'GitteHansen@itu.dk' og 'anders@itu.dk', men ikke strengene 'john@it-c.dk' og 'sidse@gmail.com'.

Bemærk: Husk at emails kan indeholde tal, bogstaver, punktummer og underscores (-)

Opgave 3.3 (5 procent)

Baggrund:

Brugerne af servicen må ikke kunne benytte JavaScript eller HTML, når de skriver indhold i servicens forms

```
function check_txt($text){
    if(ereg(DIT UDRYK, $text)){
```

```

        error("Desværre, HTML er ikke tilladt");
    }
}

```

Din Opgave:

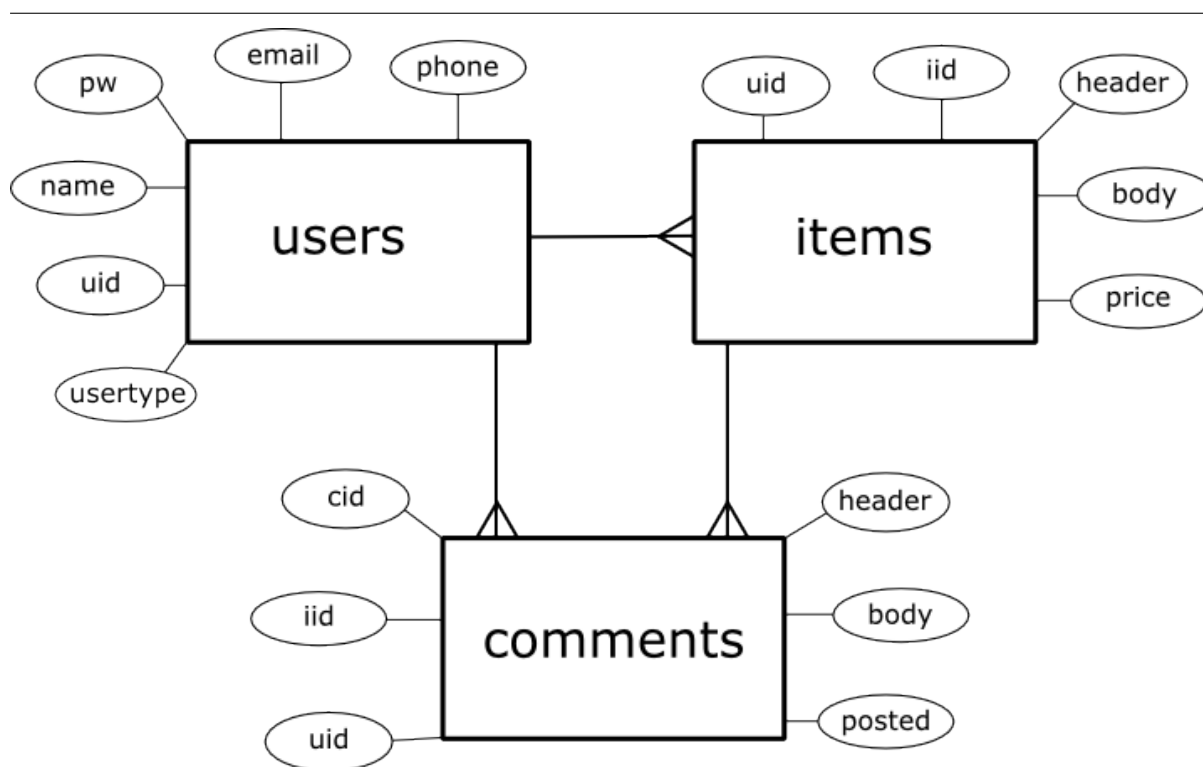
Skriv funktionen `check_text` ud fra ovenstående skabelon. Det regulære udtryk skal matche strenge der indeholder et af tegnene:

<, >, ", ' eller strengen `script`

Opgave 4 (25 procent) - SQL / Datamodel

Baggrund:

Datamodellen for systemet er angivet i Figur 2 som et ER-diagram.



Figur 2: E/R-diagram for systemet. De firkantede kasser angiver entiteterne (tabellerne), og de ovale cirkler angiver attributer (felter) tilknyttet de enkelte tabeller. En-til-mange relationer er angivet med "kragefødder".

Tabellerne `users` og `items` oprettes i MySQL med følgende SQL-kommandoer:

```
create table users (
  uid INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
  name VARCHAR(100) NOT NULL,
  pw VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  phone INT(8),
  usertype VARCHAR(100) NOT NULL
) ENGINE=InnoDB;

INSERT INTO users VALUES
('', 'John', 'shh', 'john@itu.dk', NULL, 'underviser'),
('', 'Gitte', 'shh', 'gitte@itu.dk', NULL, 'studerende'),
('', 'Sidse', 'shh', 'sidse@itu.dk', 35353535, 'forsker'),
('', 'Louise', 'shh', 'louise@itu.dk', 24252627, 'studerende');

create table items (
  iid INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
  uid INT,
  header VARCHAR(100) NOT NULL,
  body TEXT NOT NULL,
  price DOUBLE NOT NULL,
  FOREIGN KEY (uid) REFERENCES users(uid)
)ENGINE=InnoDB;

INSERT INTO items VALUES
('', 1, 'Min gamle bamse', 'Lettere udtjent, men med hjertet det rette sted', 21.50),
('', 1, 'Audi S8', 'Nysynet og med ekstra brede alu-følge', 10000),
('', 2, 'CSS for dummies (bog)', 'Ubrugt, stadig i folie', 150),
('', 3, '2 biografbilletter', 'Til valgfri forestilling i Empire Bio', 50),
('', 4, 'SonyEricsson K800i uden batteri', 'Fremstår som ny. Husk, uden batteri', 150);
```

Opgave 4.1 (5 procent)

Vis (i skemaform/tabel) resultatet af følgende SQL-kommando:

```
SELECT u.name, header, price
FROM users as u, items as i
WHERE u.uid = i.uid
AND price < 100
```


Opgave 4.2 (5 procent)

Konstruér en SQL-kommando til at oprette tabellen `comments`. Tabellen skal have følgende egenskaber:

- Feltet `cid` skal være af typen `INT` og benytte MySQL's `auto_increment` funktion.
- Feltet `uid` skal være af typen `INT`.
- Feltet `iid` skal være af typen `INT`.
- Feltet `header` skal være af typen `VARCHAR`.
- Feltet `body` skal være af typen `TEXT`.
- Feltet `posted` skal være af typen `DATETIME`.
- Ingen af felterne må indeholde værdien `NULL`.
- `uid` skal være fremmednøgle og referere til feltet `uid` i tabellen `users`.
- `iid` skal være fremmednøgle og referere til feltet `iid` i tabellen `items`.

Opgave 4.3 (5 procent)

Konstruér to `INSERT`-kommandoer, der indsætter følgende kommentarer i tabellen `comments`, du kan selv finde på overskrift (`header`) og brødtekst (`body`):

- John har kommenteret på "Min gamle bamse" den 7 maj 2007
- Sidse har kommenteret på "2 biografbilletter" den 1 januar 2008

Opgave 4.4 (5 procent)

Din Opgave:

Lav en SQL-kommando, der for hvert `item` henter overskriften (`header`), id'et (`iid`) samt antallet af kommentarer. Din SQL-kommando skal også returnere rækker uden kommentarer

Hint: Du får brug for en `LEFT JOIN` statement.

Kommandoen svarer til `index.php` i Figur 1.

Opgave 4.5 (5 procent)

Din Opgave:

Lav en SQL kommando, der henter alle kommentarer for `item`'et med `iid` 2.

For hver kommentar skal brugerens navn (`name`), kommentarens tidspunkt (`posted`), overskrift (`header`) og brødtekst (`body`) hentes. Rækkerne skal sorteres efter `posted` med de ældste først.

Opgave 5 (30 procent) - Web Service

Opgave 5.1 (7 procent)

Din Opgave:

Skriv funktionen `get_type_count`. Funktionen skal tage en streng som argument. Strengen angiver brugertypen ("forsker", "underviser" eller "studerende").

Funktionen skal først tjekke om den angivne brugertype er gyldig (altså, om den er en af de ovennævnte). Er den ikke gyldig skal funktionen returnere 0 (nul).

Er den gyldig skal funktionen, fra tabellen `users`, tælle antallet af brugere med denne brugertype, og returnere tallet.

Hint: Eksempel på funktionskald (med data fra opgave 4)

```
echo get_type_count('studerende');
//udskriver 2
echo get_type_count('rektor');
//udskriver 0
```

Du kan antage at der allerede er oprettet forbindelse til databasen.

Opgave 5.2 (7 procent)

Konstruer scriptet `insert_comment.php`. Scriptet modtager formvariablerne `iid`, `header` og `body` fra `add_comment.php`.

Husk at inkludere `utilities.php`, hvis du vil benytte scriptets funktioner.

Scriptet skal tjekke brugerinputet og kalde `error`, hvis en af værdierne er tomme, hente brugerens `uid` (`chk_cookie`) og indsætte en række i tabellen `comments`.

Til sidst skal brugeren sendes tilbage til `show_item.php` med det rigtige `iid`

Hint: Husk MySQL-funktionen `NOW()` kan benyttes til at indsætte den indeværende dato i tabellen

Opgave 5.3 (8 procent)

Konstruer scriptet `show_item.php`.

Scriptet modtager formvariablen `iid` og skal liste det givne *item* samt alle kommentarer tilknyttet.

Du skal benytte de relevante funktioner fra `utilities.php` i scriptet. Sidens `<title>` kan du sætte til "Ting til salg" eller lignende.

Det er ikke vigtigt siden ligner Figur 1, men det forventes at din markup (html) er fornuftig. Benyt eks `<h1>`, `<h2>`, `<p>` etc

Hint: SQL forespørgslen fra opgave 4.5 kan tilpasses til brug her.

Opgave 5.4 (8 procent)

1. Forklar kort, og med egne ord, en eller flere grunde til at man altid skal tjekke brugerinput.