COSC 304 - Introduction to Database Systems Assignment 7 - Creating a Web Store using PHP

This assignment practices web development using a commercial database system, Microsoft SQL Server. SQL Server supports almost all of the SQL standard including foreign keys and triggers. **The assignment is done in your project groups of 2, 3, or 4 people. If you are off campus, you must connect using VPN to access the database. Click for info. The server is myvpn.ok.ubc.ca.**

The web store that we are going to build allows customers to enter their information, chose products by putting them into a shopping cart, and then place an order.

Initial Steps

1. You can either use the cosc304.ok.ubc.ca web server that supports PHP (instructions) or setup your own local web server for development (instructions).
2. Download the starter project code files. Setup in your local development environment or on the cosc304 server.
3. Create the tables and load the sample data into your SQL Server database. The file `loaddata.php` will load the database using the `data/orderdb_sql.ddl` script. Make sure to set your own user id and password into `include/db_credentials.php` used throughout the project. You can run this file from the command line or by using the URL: `http://cosc304.ok.ubc.ca/99999999/lab7/loaddata.php` where you replace `99999999` with your student number.

Databases and Autonumber Fields

This database storing customers, orders, and products uses autonumber fields to assign a primary key value for orders. An autonumber field is an integer field which is automatically assigned by the database. The value of the counter starts at 1. When a record is added, the value of the autonumber field for the new record is set to the counter and then the counter is incremented. Thus, the values of the autonumber field for records are 1,2,3,... Autonumber fields are useful as primary keys as they are guaranteed to be unique. To create an autonumber field in a SQL Server create table statement use the `IDENTITY` keyword:

```
CREATE TABLE dummy (
   A int NOT NULL IDENTITY,
   B VARCHAR(50),
   ....
   PRIMARY KEY (A)
);
```

Helpful PHP Links

- Microsoft SQL Driver Documentation
  - sqlsrv_connect Connect to database
  - sqlsrv_close Disconnect from database
  - sqlsrv_query Execute SQL
  - sqlsrv_fetch_array Return first row from a query
- Session Documentation
  - session_start start or continue session throughout the website

    - session_destroy clear/reset all session variables
    - $_SESSION array
- isset Check if variables are inuse

Question 1 (10 marks)

Modify the **`listorder.php`** so that it lists all orders currently in the database. You must list all orders and the products of those orders.

Details:

1. Sample output
2. When you upload your site to the server in the folder `public_html/lab7` then your URL will be: `http://cosc304.ok.ubc.ca/(yourUnivId)/lab7/shop.html`.
   For example, my web site is at: `http://cosc304.ok.ubc.ca/rlawrenc/lab7_php/shop.html`.
3. The main shop page is `shop.html`. Feel free to change it to your shop name and style!
4. Your output does not have to look exactly like the sample (feel free to make it look better!).
5. A good way to get started with `listorder.php` is to start with the sample PHP code posted (QuerySQLServer.php) and modify it for this particular query.

Marking Guide:

- **+1 marks** - for SQL Server connection information and making a successful connection
- **+3 marks** - for displaying order summary information for each order in a table
- **+4 marks** - for displaying items in each order in a table (must use prepared statement)
- **+1 mark** - for formatting currency values correctly (e.g. $91.70)
- **+1 mark** - for closing connection

Question 2 (30 marks)

Build a simple web site that allows users to search for products by name, put products in their shopping cart, and place an order by checking out the items in their shopping cart. Starter code is provided. Fill in a few of the JSP files to get the application to work. Here are the steps you should do to get started:

1. Use the template code downloaded and setup in Question 1. Summary of files:
    - **`listprod.php`** - lists all products. **TODO: fill-in your own code (10 marks)**
    - **`addcart.php`** - adds an item to the cart (stored using session variable). No changes needed.
    - **`showcart.php`** - displays the items in the cart. No changes needed.
    - **`checkout.php`** - page to start the checkout. No changes needed.
    - **`order.php`** - store a checked-out order to database. **TODO: fill-in your own code (20 marks)**
2. Take a look at the sample web site which is available at http://cosc304.ok.ubc.ca/rlawrenc/lab7_php/shop.html.
3. Start by editing the PHP file called **`listprod.php`**. This file is called from `shop.html` when the user begins to shop. The file allows a customer to search for products by name. If a customer enters "er", then the query should be: `productName LIKE '%er%'`.
4. Start off with just being able to list products by name. Inside `listprod.php` is a form whose GET method calls `listprod.php` itself. When a user submits the form, the URL passed to `listprod.php` will contain a parameter `productName`. Based on this parameter, construct your query. Start with the template code and then add the required code to connect to the database and list the products.
5. The file `listprod.php` also allows users to add items to their cart. This is accomplished by having a link beside

each item. When the user clicks on the link, another page called **`addcart.php`** is called with information on the product to add.

6. The file `addcart.php` expects the following parameters: `addcart.php?id=(productId)&name=(productName)&price=(productPrice)`. You must make sure that you create the appropriate links when listing your products.
7. `addcart.php` calls another file that maintains a record of the shopping cart over a user's session. This file is `showcart.php`.
8. When the user wants to check-out, they must enter customer information. The file `checkout.php` prompts the user for a customer id and passes that information onto the PHP file `order.php`.
9. The other file you must write is `order.php`. This file must save an order and all its products to the database as long as a valid customer id was entered.
10. Make sure to list the order id and all items as shown in the example.

Marking Guide (for listprod.php): (10 marks total)

- **+2 marks** - for using product name parameter to filter products shown (must handle case where nothing is provided in which case all products are shown)
- **+1 mark** - for using PreparedStatements
- **+3 marks** - for displaying table of products
- **+3 marks** - for building web link URL to allow products to be added to the cart
- **+1 mark** - for closing connection

Marking Guide (for order.php): (20 marks total)

- **+3 marks** - for validating that the customer id is a number and the customer id exists in the database. Display an error if customer id is invalid.
- **+1 mark** - for showing error message if shopping cart is empty
- **+4 marks** - for inserting into ordersummary table and retrieving auto-generated id
- **+6 marks** - for traversing list of products and storing each ordered product in the orderproduct table
- **+2 marks** - for updating total amount for the order in OrderSummary table
- **+2 marks** - for displaying the order information including all ordered items
- **+1 mark** - for clearing the shopping cart (sessional variable) after order has been successfully placed
- **+1 mark** - for closing connection

Bonus Marks

Up to 10 bonus marks can be received by going beyond the basic assignment requirements:

- **+5 marks** - for allowing a user to remove items from their shopping cart and to change the quantity of items ordered when viewing their cart.
- **+5 marks** - for validating a customer's password when they try to place an order.
- **+5 marks** - your site runs on cosc304.ok.ubc.ca or another server not just on your local development machine
- **Up to +5 marks** - for improving the looks of the site such as:
  - **+2 marks** - for a page header with links to product page, list order, and shopping cart
  - **+3 marks** - for formatting product listing page to include better formatting as well as filter by category
  - **+3 marks** - for improved formatting of cart page
- Other bonus marks may be possible if discussed with the TA/instructor.

If you want to be eligible for bonus marks, please note that on your assignment and explain what you did to deserve the extra marks. An example web site with improved features is available.

Deliverables:

1. Option #1: Demonstrate your working site to the TA in the lab and get +2 bonus marks. No submission on Canvas is required.
2. Option #2: Submit in a single zip file all your source code using Canvas. This can be done by exporting your project. Submit all your files, but the files you must change are: `listprod.php`, `listorder.php` and `order.php`.
3. You do NOT have to get the code uploaded and running on the web server cosc304.ok.ubc.ca to complete the assignment. However, if you do, you can submit the URL on the server for the TA to test your assignment.
4. If you work in a group, only one person needs to submit the assignment. Put all partner's names and student numbers on the submission.

Home
