BIRZEIT UNIVERSITY

Computer Science Department

Web Application and Technologies (COMP 334)

Second Semester 2023/2024

Assignment 2: PHP ***Due Date 18/05/2024***

Objectives

To demonstrate the ability to create a data-driven PHP page and to use super global arrays.
To practice SQL processing through PHP scripts.
Notes:

- You must define your database connection details in a separate file called "dbconfig.in.php", it should **define** the required variables used to create a connection and ***create a PDO object*****. The database connection must be of type PDO.**
- You should always use **ONLY** prepared statements with named binding parameters for any database SQL.

Overview

For this assignment, you will write a web application using PHP and a database that allows users to add, view, update, and delete products on the e-clothing web store you have started in the first assignment. The main script of your application is called "*products.php*" which dynamically generates an HTML page, as shown in Figure 1 below. The page has three parts: the first is a link to **Add** a new Product, the second is a link that allows the user to perform a **filter search**, and finally, the **products' table displays** the search result. When the script is called for the first time the table should display all the products' records.
The search form consists of a text box and a list of radio buttons. Notice the placeholder text in the search box and a Filter button that sends the user input to the "*products.php*" script. Set the form's method attribute to POST.
Create a table of products that looks like that shown in Figure 1.
The action buttons in each row are a series of <button> containers with a link to the appropriate PHP script and an image within the button. See the edit and delete actions below.

Actions:

Search: allows users to search for products. The user can select which fields to perform the filter search by Product Name, which regenerates the products' table with records matching the search criteria, or by Category or Price. For instance, if the user enters 80 in the search text field, selects price from the radio button, and then clicks the filter button. The user inputs will be sent to the "*products.php*" script, which returns an updated page version by displaying only the products

whose prices are *greater than or equal to* 80. The search can combine Price and Category, or Name and category, or Name or Price. A name search can match part of the name.

Figure 1: prodcuts.php

Product details are retrieved from the database, and **the data should be fetched to a Product Object.** So, you have to create a Product Class that holds the product information: product ID, product name, category, description, price, rating, and product image name. In addition to the constructor and the necessary setters and getters methods, the class has the following methods: *displayInTable* returns product details as an HTML table row, and each table cell holds the product's details: product ID, product name, category, description, price, and image. The Product ID should be hyperlinked, and the anchor element should refer to the "view.php" script, as fully described in the **view** action below. The second method, displayProdcutPage, generates a <main> HTML element to display the product page, as shown in Figure 2 below.

View: The Product ID should be hyperlinked, and the anchor element should refer to the "view.php" script, so when clicked, send a GET request. So, combine the query string with anchor tags so the "view.php" script receives as input a query string consisting of the product ID that specifies which product to display. The"view.php" script dynamically creates an HTML, as shown in Figure 2. Product details are retrieved from the database. If an invalid ID has been sent, an HMTL page with an error message should be displayed.

Product ID: 210, Super Non-Iron Striped Slim-Fit

- Price: 85
- Category: Formal Shirts
- Rating: 4/5

Description:

Natural 100% cotton with no ironing required. A high-quality shirt designed with attention to detail.

- 100% cotton fabric made from fine-count, double-ply, ultra-long fiber cotton.
- Supple and glossy Pinpoint Oxford fabric.
- Non-iron natural 100% cotton.
- Manufactured to strict quality standards, from fabric production to sewing.
- Low-maintenance non-iron shirt.
- Collar height and shape designed to look good with a necktie.
- Shoulders designed for easy arm movement.
- Pairs well with any suit and tie.

Figure 2: prodcut's Detail Page

- **Add:** When the user clicks the add link, the request will be sent to the “add.php” script. The script is a self-reference script; if no data has been sent, a form is generated, as shown in Figure 3. Otherwise, the data is retrieved from the form and then stored in the database. The form's method attribute should be set to POST and action to “add.php”. The product photo should be uploaded to a folder called “images”, only accept an image of type “jpeg”, and rename it to be the same as the prodcut ID. For instance, if you have a product with an ID of ‘112’, the file name should be ‘112.jpeg’. Also, you should save the file name in the database.

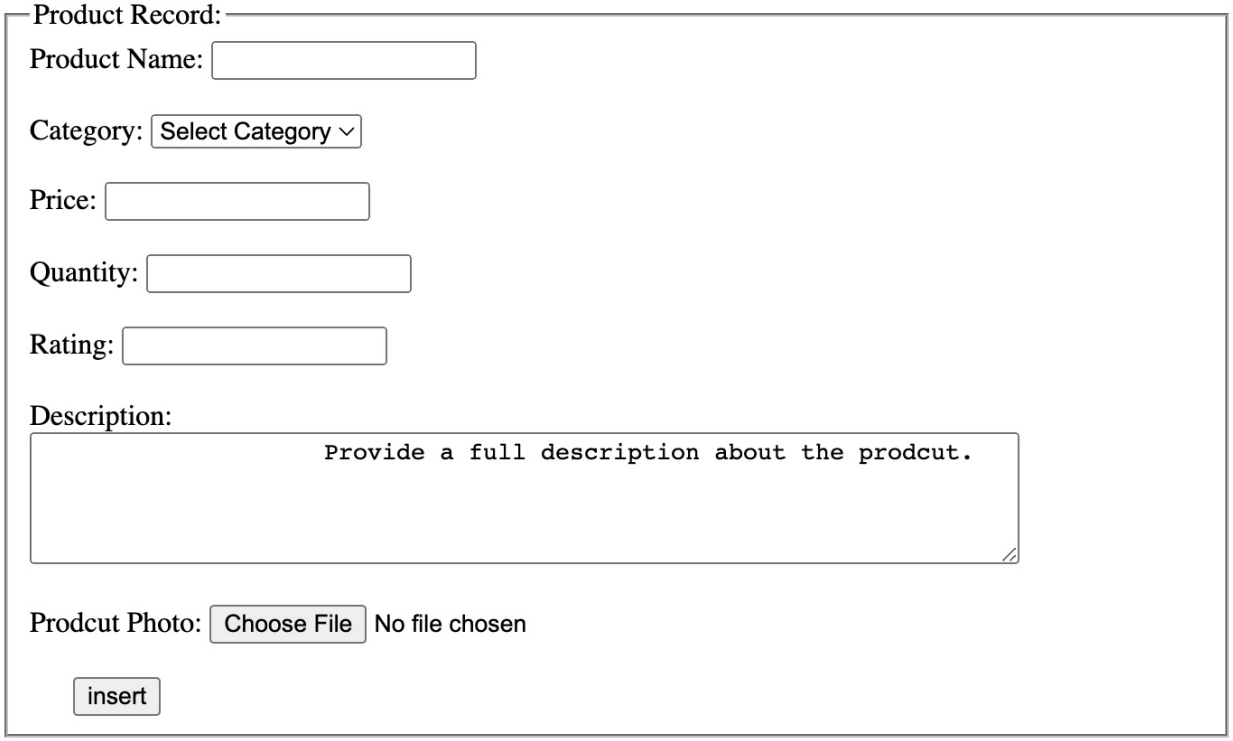

Figure 3: Add Product Page

- **Delete** when clicked a get request with the product ID sent to a script “delete.php”. Which deletes the product Record from the database. Also, as discussed in the View action above, you need to combine the query string with anchor tags.
- **Edit** allows the user to update editable product details: price, quantity, description, and product image when clicking a get request with the product ID sent to a script “edit.php.” Also, as discussed in the View action above, you need to combine the query string with anchor tags. The edit script returns an HTML page with a form filled with the product’s details, which are retrieved from the database; the user should be able to change only the editable fields. Then, the user sends the new updated product details to the script “edit.php,” which updates the product's fields in the database. The page generated by the “edit.php” script is the same as shown in Figure 4.

Product Record:

Product ID: 210

Product Name: Super Non-Iron Striped

Category: Formal Shirt

Price: 85

Quantity: 15

Rating: 4

Description:

```
Natural 100% cotton with no ironing required. A high-quality shirt designed with
attention to detail.
  100% cotton fabric made from fine-count, double-ply, ultra-long fiber cotton.
  Supple and glossy Pinpoint Oxford fabric.
  Manufactured to strict quality standards, from fabric production to sewing.
  Low-maintenance non-iron shirt.
  Collar height and shape designed to look good with a necktie.
  Shoulders designed for easy arm movement.
  Pairs well with any suit and tie.
```

Prodcut Photo: Choose File No file chosen

update

General Requirements for all pages.

- All pages should include a header with the store name, logo, and navigation for other site pages described above.
- All pages should include a footer with the following information: the last update date, store address, customer support information: telephone number, email address, and a link to the Contact Us page. The footer information should always appear at the bottom of each page.
- In your web pages, you should use semantic tags like <article>, <nav>, or <footer> (for example), and some parts make sense to wrap inside a tag such as <section> or <figure> instead of generic HTML tags like .
- You must use relative addresses for all links to the documents and resources within your site.

Placement of Assignment Files

- You must submit your files to CS Host.
- In the “public_html” folder, create a sub-folder called *assTwo, and then upload all the assignments* files to it.
- You must update your home page (index.htm) file by *adding a link to the assignment Two main page*.
- You must create a database on the server named as “clothingStore” and load it with at least 10 records for testing. Also, the database Schema should be exported as SQL, the file name should be"students_yourNumber.sql".
- **You must compress (Zipped) all the assignment’s two files, including the SQL file, into a file named assTwo-stID.zip and submit it to ITC before the due date.**

 You must submit your work before the due date *which is on 18/05/2024 at 22:00 PM*, by uploading it to the **CS host and submitting it to the ITC**.