BIRZEIT UNIVERSITY

Computer Science Department

Web Application and Technologies (COMP 334)

First Semester 2023/2024

Project: PHP Script, CSS, and HTML Forms

Due Date 22/01/2024, see submission instructions below.

You are developing a website for an online store. The store is marketing Palestinian souvenirs for international customers, products such as but not limited to: handcraft (such as منحوتات ، مطرزات و الفخار), ceramic (الخزف و الزجاج الملون و الفخار), natural product (صابون نابلسي، طينة البحر الميت), and food (زيت الزيتون، زعتر، دبس، قطين). The website should allow customers to search for products and buy them online. The website should have the following functionalities:

- Search through the stores' inventory of products
- Place an order for products.
- Receive an acknowledgment message that confirms the order has been placed.
- Allow the store manager to manage inventory, by adding products, updating product details, etc.
- Customer Registration

***Note:** the HTML pages (including forms) should be dynamically generated using PHP scripts.*

User portal types

You have two groups of users:

- Customer: can do the following functions
  - Customer Registration
  - Product Search
  - Place an order.
  - View an Order.
  - Add/remove an item to/from an order
- Employee: can do the following functions.
  - Process an Order.
- View an Order.
  - Manage inventory, by adding products, and updating product details.

Add Product

Store employees should be able to enter a new product or update its details. These are the minimum information that each product should have.

Product Name, Product Description, Price (numeric data only), and category that should be selected from a pre-defined list, {new arrival, on sale, featured, high demand, normal} and displayed in a combo box to the user, the default category is normal for simplicity a product belongs only to one category. Quantity which should be numeric.

Each product should have the following:

- Name
- Brief Description (Enable the text box to accept multiple lines of text)
- Category
- Price
- Size
- Remarks (Enable the text box to accept multiple lines of text)
- Product ID which is a 10-digit number given, is generated by the system. When the user completes entering the product details the system must display a confirmation message that confirms the product has been successfully added to the database and displays the product ID.
- At least **one** picture for marketing, you need to add a link to allow the user to upload files. I might suggest that you have to upload at least **one** picture for each product. You can save the file name in the database. The filename should be like that itemIDimgSequenceNo, for instance, if you have an item its id is 12345, and then the first image file will be *item12345img1.gif*. The file itself could be saved in a folder named as *itemsImages.*

Update Product Quantity

Store employees should be able to update the quantity of any product, so the systems should allow the employee to search for a product by name, part of its name, or product ID. The system displays a list of products as described in the search list below. The user clicks on the product ID, and the systems display a form that asks for the new quantity to be added, so the product quantity will be summed up to the current quantity.

Customer Registration:

For a customer to place an order the customer should have an account in the system. The customer creates an account through the following registration steps:

a) The customer fills out the registration form by providing the following information.

Customer info. (all the information is required.)

- Name
- Address, {Flat/House No, Street, City, Country}
- Date of Birth
- ID number {رقم الهوية}
- E-mail address
- Telephone
- Credit card details: The customer should enter his/her Credit card details: number, expiration date, name, and bank issued it.

All the above fields should be filled once this step is validated, and then the customer can move to the next step. *You must use the session management to implement this functionality.*

b) The customer can move to the next step, which is creating an e-account. Your site should display an E-account form that allows the customer to enter a username which should be between 6-13 characters. Also, the user should enter the password and password confirmation. The valid password should be between 8-12 characters.

c) The last step is the confirmation step, the system displays all the information entered by the user in the previous two steps in a form. The form has a confirm button that sends the complete form to the server. Then the server stores the data to the database and replies to the customer with a confirmation message. The confirmation message should have the following information. The customer ID which is a 10-digit number, (customer ID will be generated by the system), and a link to the login page.

User login clicking the link open login form which asks the user to enter his username and password.

User Logout clicking the link logs the user outside the system and destroys the session.

Login Use Case Specification

On your home page on the side navigation bar, you should add the following links.

Login for returning users, clicking the link opens the login form.

To register as a new user, clicking on the link starts the user registration use case as described in user registration, above.

The user logs in to the system and the User menu is generated based on the user type, which could be a customer or an employee.

Product Search

The system should allow the user (any site visitors, either have an account or not) to do a filter search. So, the search functionality should be available for all site visitors (by product name, and price).

<u>Search by name and Price Search</u>

The site visitor should be able to do a filter search. The system should display a search form that allows the user to enter a price range between two values, and a product name either full name or part of the name. So, if the user leaves the name field empty the search will be done based on the prince range. If the user enters both the price range and name, then the system will search for the products that match the price and name criteria.

The search result List Specification:

1. The returned list should be displayed within a table the first column is a checked box that could be used by the user for a shortlist. So, the header of the 1st column should be a button when clicked the table is updated to display only the checked items.
2. The second column in the table has a reference number which is unique within the database you can consider it as a primary key. It is hyperlinked when clicked a detailed page about the product is displayed as given below.
3. The Third column should be price and the fourth is the category. The Table titled should be clickable, when the user clicks on it the list is sorted according to that column. Use the cookies to remember the list order. The fifth column should display the product quantity.
4. You should use a CSS class to distinguish the presentation of the products in the table based on their category. So, you must create a CSS class for each product category and write CSS rules to change the background and font style.

<u>Product Details Page specification:</u>

1. Product details should consist of an image of the product that is displayed on the left, and the product description should be displayed to the right. You should use positioning and floating **CSS** properties to arrange the product display.
2. A link to add the product to an order should appear below the product details box. See the place order specification below for details.

<u>place order specification:</u>

1. The user should be logged on to place an order, therefore, if the user is not logging into the system. The login form should be displayed first. After a successful login, the item can be added to the order. *After a useful login the system should return the user to where s/he left, i.e s/he should not start over again.*
2. The order should display the items in a table in the following order: product id, title, price, quantity, remove link, which removes the item, from the list and recalculates the total order amount, then the list should be re-displayed on the same window. Note: you can make the quantity ZERO for the item, but you should leave it on the basket.
3. The form should have a recalculated link that updates the order. Since the user can update the quantities of the product.
4. Check out the link, the server sends the checkout form. The checkout form should display the following details, customer details, shipping address, credit card details the item list, and total the amount.

5. The checkout form is submitted to the server when the “confirm” link is clicked. The server stores the order on the database and then displays a message to the user thanking him/her for purchasing and informing him/her about the order ID. The order ID should be hyperlinked, by clicking on it the order details are displayed on a new window.
6. Once the customer places the order, the order is stored in the database. The order’s initial status of the order is waiting for processing.

View an Order.

Customers and Employees should be able to view a list of orders, the list should be as follows:

- The customer list should be displayed in a table as follows: the list should be sorted to the newest order at the top of the list, and the first column should be an order ID that is hyperlinked and allows the user to view the order in a sperate window or tab, 2nd column order date, 3rd column order total amount, 4th column order status {waiting for processing, shipped}
- The employee list should be displayed in a table as follows: the list should be sorted with the orders waiting for processing at the top of the list the oldest displayed first, the first column should be an order number that is hyperlinked and allows the employee to view the order in a sperate window or tab, to update the order status/ or view the order, 2nd column order date, 3rd column order total amount, 4th column order status {waiting for processing, shipped}

***CSS**: requirements* you should have **different** styles for orders based on order status, the waiting for processing should be displayed using bold font weight, using the appropriate class to do so.

Process an Order

Only an employee can process an order, from the order list described above, the employee can view the order, and the system displays the order details in a form in which all the fields are disabled, i.e. not editable, **except** the status field which could be changed from ‘waiting for processing’ to shipped and an input field to allows the user to enter the shipping date and time. NOTE, it is not allowed to change the status of a shipped order. So shipped orders can only be viewed.

CSS Requirements:

All your web pages and CSS code must validate correctly, and all your formatting should be done using CSS, using the appropriate CSS selector and all CSS rules must be stored in one external CSS file. The layout of your pages should be similar to the layout shown in Figure 1 below. You must use a flex container to create the page layout.

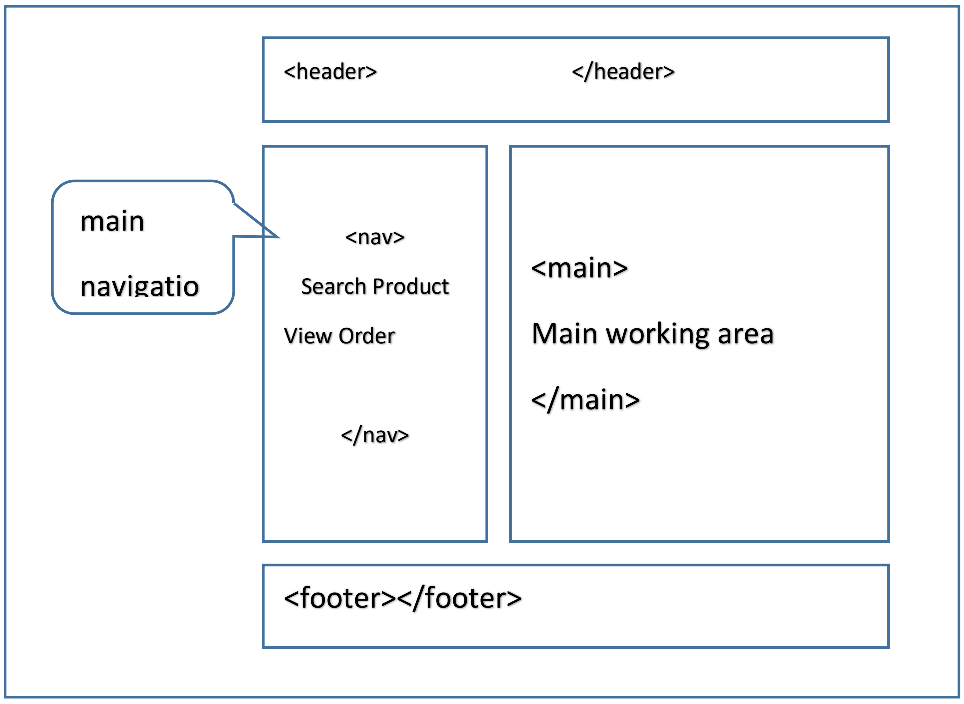

Figure 1: Page Layout

1. Each **page** should contain 4 sections called **"head"**, **"nav" "main" and "footer"**. The header section should extend horizontally across the top of the entire page. The navigation section and the display section should run vertically below the header section. The navigation section should be on the left side and be 15% of the width of the screen. The display section should consume the remainder of the screen.
2. The **header section** should contain:
    - The e-store name
    - store logo
    - About us page link.
    - User Profile (user name and username)
    - Shopping basket, which shows the current order.
    - And register/ login logout/ link.
3. **Footer section**: smaller logo and copy write text, such as an address, contact email, and telephone number for customer support and a link for the Contact Us page.

4. The **main section** changed from one page to another depending on the task being performed, i.e., all pages have the same header, navigation, and footer sections and differ in the content of the main section. For instance, when the user selects search the search form is displayed, and the table result appears underneath. In the navigation section the activated link, if the user selects search the search link is activated and should be displayed in a different style in the navigation section. When the user selects the view order link, the link is activated by changing its style in the

navigation section and loading the appropriate page to the main section. The header, navigation, and footer sections should not change. *The links in the navigation section should have a style different than other links on the site*. The **main section** should be used to display the results of the links chosen from the navigation section. Upon initial load, this section should be loaded with promotional materials such as new arrivals products and offers of the day.

CSS REQUIREMENTS

- Use only external CSS files, NO embedded or inline CSS is allowed.
- All the HTML files should have a consistent presentation format and layout. Page Layout should be like Figure 1 and consist of a navigation section on the left, a header section on the top, a footer section on the button, and the main section on the center.
- Make sure you use the CSS rules that are given to you in the lecture ONLY!
- It must use the Flex property to implement the side-by-side layouts unless specified otherwise.
- Use the correct element types, for instance, nav for the navigation bar, etc. You must *use semantic elements*. General elements such as div are allowed **unless** there is no available semantic element. Use <em> instead of <i>, <strong> instead of <b>, and provide style.
- You must use classes and ids in your solution in the right way. For example, the system should mark the input field with a special background color if the user forgets to enter the required field; also, it should display an appropriate error message near that error input field. So, create a class called “error” to give a style error message that will be displayed to the user when forget to fill mandatory field.
- In the user forms, all fields that are required should have a special background color. So, use CSS classes for this option. You should provide a style for the **required** fields and must be consistent with all forms. All mandatory inputs should be clearly marked. Even *if all the input fields in a form are required, they must be clear for the user*. In other words, you should write appropriate CSS rules not to depend on the browser to deal with required fields.
- Change default font properties in 4 locations, headers, table, .., etc
- Change box model properties (border, padding, margin) in four different locations on your website
- You must use the figure element to display the images and change the position properties to display the images and the captions.
- All input text should change the background on the focus.
- All **input labels** should be surrounded by a border in a different color than the text label color.
- The **table header** should have a different background and text color than the table data. Header data should be centralized.

- The table result should have a border and consume 100% width of the containing element.
- **Links outside** your website should have a different color schema (color for visited, non-visited, and hovering) than those within your website. The difference should be clear enough to be disguised. Also, they should be different in text decoration.
- The **table rows** should have a different background for even rows than for odd rows.
- **Lists:** unordered list select an image to use it. When an unordered list appears within another unordered list, use a different image. Ordered lists use Arabic digits for the top level and alphabetical for the nested level. Note: write the CSS rules even if you did not use any list on your site.
- **Submission button**: use images (metaphors) that indicate the task, provide a style to add a border, change the background color upon click, and change the default text color.
- *You must submit the final style rules, all the classes must be used within the HTML pages. All CSS files must be cleaned from all temporary and not-in-use classes.*
- *Any work taken from the internet must be documented by adding a reference to the source. The code {HTML, CSS, and PHP} taken from the internet or somewhere else should not exceed more than 10% of your total work and **should not be a complete functionality**.*

Submission instructions

The Deadline for submitting this task is Monday 22/01/2024. ***At 14:00 at the CShost***, the server will be closed on Monday 22/01/2024 at 15:00. *In addition to your submission to the CShost you have to send me your project to ITC by the deadline which is* Monday 22/01/2024. What you have to submit to ITC: your project files should be compressed, and the compressed file should be extracted to a Folder named stdID (where std is your name and ID is your student ID). Make sure when you extract the compressed file to the localhost and type the following URL http://localhost/webprojects/stdID to the browser address bar the index page of your project will be loaded correctly, and all the links are working fine.

The folder should have the following: -

- All the PHP scripts, HTML files, CSS files, images, ..etc.
- The database Schema is exported as SQL, the file name should be"dbschema_yourNumber.sql".
- Index.html

The index page should has:

1. Name and Student ID.
2. A link to the Main/index page of your project
3. You should create two users one customer and an employee and provide their details, for testing: username and password.
4. You should load your database with some products for testing