PROCEDURE FOR SOFTWARE WORK AND DEVELOPMENT

Example of a project assignment: Online shop

Note: The topic can be changed if it meets the requirements. This is just an example.

The goal of this assignment is to create web app which represents online shop for showing and shopping items from 3 categories:

- Books,
- Music,
- Videos

The assignment should be done through these sub tasks:

Task 1:

- Create classes that represent items that can be purchased in an online store (classes should meet the OOP specification)
    - Base class Item or Article needs to have these elements: Id, Title, Publish Date, Short Description, Price.
    - Other classes should be derived from Item or Article class and have additional fields:
        - Class Book also has Author Name (can be more than 1 author), Name of Publishing House, Number of Pages,
        - Class Music also has Author Name (can be more than 1 author), Name of Publishing House, Duration in Seconds, Number of CDs (some albums can be on 1 or more CDs),
        - Class Video also has Director Name, List of Main Actors, Duration in Minutes, Number of DVDs (some films come in multi-DVD packages)
- Create 3 DAO classes to work with the database.

Task 2:

- Create class that represents a registered Buyer with these attributes: Firstname, Lastname, Address, Email, Phone, Password (password is used in log in process), Role
- Create DAO class to work with the database

Task 3:

- Create class that represents current Cart of a registered Buyer which contains list of items in cart, their quantity, total price.
- Create DAO class to work with the database

Task 4:

Create View page(s) for showing current available items shown by categories. Somewhere on the View page should be an option for choosing the category which is currently shown (can be link or drop-down). All items should be sorted by names in ascending order.

A maximum of 10 items can be displayed on the page, so it's necessary to make the pagination functionality:

- The user is displayed current page number of total pages (*example: Page 1 of 3*),
- The user is displayed links to next and previous page,
- The user is displayed total number of items in category (*example: Total 123 books*).

Work with items was partially done in task 1 and task 4.

Note: Every item should have their own image, preferably every different item has different image, but to ease things up, every item in same category has the same image.

Task 5:

- Create page with log – in form for registered users (buyers). User enters email as username and password, also page should contain link for registration if the user (buyer) isn't already registered (*second part of Task 5*). If the user entered wrong email or password, redirect to log – in page with the error message, otherwise user is redirected to items page.
- Create page for user registration. On the page, user should enter Firstname, Lastname, Email, Phone, Address and desired Password. After entering the data, it is checked whether all fields are valid and whether there's already existing user with the same Email address. If there's a user with the same Email address, return an error message. If all fields are valid, registered user is redirected to the items page.

Task 6:

Complete the page(s) from *Task 4* with following:

- If the user currently looking at the items and it's not logged – in., show message "*You are not logged in!*" somewhere on the page. If the user is logged – in, on the same place show Firstnale and Lastname (*example: You are logged in as Firstname Lastname*).
- Set the option for logout next to the message from previous sub task.
- If the user is logged in, show option to add items in cart (*shopping*). Beside or under every item set the link for adding item in cart. If the item is already in the cart, increase its quantity by 1. After inserting the item into the cart, write a message that the item has been successfully added.
- If the user is not logged in, disable the shopping functionality. Beside item there's no link for adding item in cart.
- Set the data about current cart, show how many items are currently in the cart and what is total price. Set the option for checkout, when the user clicks on checkout link show all content from cart (*page with cart is defined Task 7*).

Task 7:

Create a page for viewing current content in the cart and confirming the purchase. The page lists each purchased item along with the quantity. Next to the item, there's an option to remove the item from the current cart. There's also an option to delete the entire contents of the cart. The page should also have a link to continue shopping, which takes the user back to the item page. The last option on this page is a purchase confirmation, which empties the cart and displays the invoice page with the purchased items. On this new page, at the bottom, there should be an option for the customer to checkout or continue shopping.

Task 8:

Add the functionality of working with the cart so that when the user logs out without first emptying the cart or completing the checkout, it saves the contents of the cart.

Clarifications

If something is not clear enough or explained, make some reasonable assumptions when solving tasks. It’s important that all functionalities are satisfied, and the problem can always be solved in many different ways.

Login functionality can be implemented so that when the user is successfully logged in, his object is placed in session. Cart should also be in session. At the logout, that object should delete from session.