**Open Point Webshop**

# Task 6.2: Select Options

▶ After selecting a "Buy Now" link or button, display a new page with additional

product options (e.g., customization options or recommended accessories)

plus further options such as "Shipping Method", "Payment Method", "Gift Box",

etc. For this, use radio buttons and check boxes.

# Task 6.3: Shipping Address

▶ Transmit the selected options to a new page with a form for entering the

client's name, e-mail, and shipping address. Use different form elements, e.g.,

a selection list for different countries or a textarea element for a comment.

# Task 6.4: Confirmation

▶ Transmit all the entered data to a confirmation page, which informs the client

about the purchased product, the options, and shipping address. Send

automatic e-mails to both the client's and the shop's official addresses.

# Task 6.5: Refactoring and Code Beautification

▶ If you have not done so yet

▶ Make sure you are following the usual code conventions and have

sufficient comments for others to understand your code.

▶ Simplify your code by extracting appropriate PHP functions. For example,

write a function which takes a menu name and a hyperlink as input, and

then produces the necessary HTML code for the navigation menu. Or you

may want to define a function to generate the entire navigation menu or

the list of products. Or…

▶ Try to make your code as compact and flexible as possible, e.g. by strictly

separating PHP and HTML code.

# Task 7.1: Purchase Confirmation

▶ At the end of the ordering process, display a JavaScript confirmation box to

inform the user that he/she is about to enter a binding contract of purchase.

Don't proceed in case the user clicks "Cancel".

# Task 7.2: Form Validation

▶ Install corresponding functions to validate the entered data, i.e. check that no

required field remains empty, that phone numbers are real phone numbers,

ZIP codes are real ZIP codes, and e-mail addresses are real e-mail addresses.

Don't proceed with invalid data.

▶ (Note: For more complex validations, regular expressions will be introduced in

Section 15 of the Lecture)

# Task 7.3: DHTML

▶ Make your page a bit more interactive and dynamic using DHTML techniques.

For example, highlight a menu item when hit by the mouse pointer, add a

system clock, install collapsing product information or collapsing sub-menus,

# Task 9.1: OO Shopping Cart

▶ Refactor the shopping cart using OOP. Define a class ShoppingCart with

methods for adding items, removing items, emptying the cart, calculating the

total price, displaying the cart as HTML, etc. For this, you may want/need to

introduce additional classes, e.g., for individual shopping cart items. Store the

shopping cart object in a PHP session variable to pass it from one HTTP

request to another.

# Task 9.2 : OO Product List

▶ Define classes for single products and for the entire product catalog. Use it to

replace any existing array-based implementation. The goal is to make your

implementation more flexible with respect to the upcoming data base

integration.

# Task 11.1: Web Service Integration

▶ Pick a REST of SOAP service of your choice and integrate it into your

shop.

# Task 12.1: AJAX

▶ Add some AJAX-based effects to your Web shop to speed up the user

interaction. You could for example add a product search page, which shows

you the resulting product list as soon as you start typing the query. Or you

could load and show a detailed product description as soon as you click on or

move your mouse over a product.

# Task 12.2: HTML Templates

▶ Using templates, try to reduce the amount of HTML code in your PHP scripts.

For this, you may want to try out one of the existing HTML template systems

(e.g. vLib or Smarty).

# Task 12.3: Regular Expressions

▶ Use regular expressions to simplify any complex string manipulations in your

code. In particular, try to improve the validation of the user's input data, in

which case you may need to use the JavaScript (client-side) or PHP (server-side)

functionalities for regular expressions.