



### Centre for Continuing Education

#### 420-P75-AB PHP PROGRAMMING

#### Test 2

Date: March 28th, 2019

This test is marked on 50. It is worth 25% of the course.

Time: 4 hours

Good Luck!

Your task for this test will be to implement a todo list based on the html pages provided. Note that some page offer different options are to be made available depending on the logged in status of a user.

Any possible styling class or options needed for the layout (either for error or based on whether a user is logged in or not) are already included in at least one of the html files provided.

Provided on Léa are the files that you are to use for the test. Import the sql file using phpMyAdmin in your database *IPD16*. When submitting your test you should be using username *ipd* and password *ipdipd*. Additional files provided include, HTML, CSS and images used for the layout of the website

Full marks require you to use either a framework or meekro/twig to complete this exam.

Your exam should be located at url <http://localhost:81/todo> (so your todo folder should be in htdocs !)

### Pages Listed in Menu

#### Home - list.htm

A table listing all the tasks to get done or those that are completed

There are 4 possible looks to the rows in this table. All of which are represented twice in list.html

Any user is allowed to complete a task

When a task is marked as completed it should be ~~struck through~~

Once completed the "*completed*" button should no longer appear for that task.

Only a user who is logged in is allowed to delete a task (removed from the database)

*Consider completing this conditional statement after you've done the login*

When "Completed" is clicked

The task gets updates in the database and you should be send back to the home page

When "Delete" is clicked - *only if the user is logged in*

The task gets removed from the database

#### Form - form.htm

A basic form that allows anyone to create a new task

The name field is *required* and the description is *optional*

*required means it must have some information filled in*

*optional means it is not a requirement that the user fill in this form element*

Perform the proper validation base one the required/optional fields

The current html has the error class and message already displayed.

Display this class and error message to the user only when there is an actual error

#### Login - login.htm

A form the will login the user based on the entries in the database

Perform the proper validation base on the required/optional fields

Display error to user only when there is an actual error

*consider using the class and message from the form.html*

Sessions variables should be used to track the user status

*Is this script polite and saying Hello to anyone?*

#### Logout

Logs out the current user

## Forms

All your forms should use the POST method.

Don't forget to set the names and values to your form elements.

Perform the proper validation base on the required/optional fields.

Only display error to user only when there is an actual error.

## Database

### **todo\_users**

listing of the users that are allowed to log in

Queen Elizabeth, Prince Charles and Prince William use the password **1234**

All the others use the password **123**

### **todo\_tasks**

listing of all the tasks

*completed* - should contain either 0 or 1 to identify if a task has been completed (*think boolean*)

*ip\_address* - should be updated to hold the current IP address of whoever created the task  
(*maybe PHP probably has a function to do this?*)

Save all files, zip the folder and submit it to Lea in the Assignment section for Test 2.

(should your zipped folder be too large to submit via léa, remove the vendor folder and try to resubmit)