Exercise 4 – Laravel

Weighting: 2%

In this exercise we are going to write and modify existing simple Laravel applications.

Task 1 – Create a Laravel project

Create a blank Laravel project. Make sure it runs by displaying the Laravel Welcome page.

Hint: following lecture video.

Task 2 – Greetings in Laravel

Implement the Greetings example (from Week 3) in Laravel.

Hint: follow the lecture video.

Task 3 – Implement foreach $_GET in Laravel

Your solution to this task should be placed in a directory called task3.

Implement the foreach $_Get example in Laravel. This foreach $_GET implementation should do what is shown in Week 3 lecture, in addition, it should do the following:

1. Display alternating row colours in the displayed table.
2. If there is no URL variable, then display “No URL variables”.

Hint: follow the lecture video.

Task 4 – Greetings template inheritance

Your solution to this task should be placed in a directory called task4.

Implement Greetings example by using template inheritance.

Hint: follow the lecture video.

Note for Peer Reviewer

For Tasks 2, 3, and 4 ensure your reviewee explain the flow of execution; from form generation to submission, processing, and displaying results. Ensure your reviewee have explained and understood the following:

- Routes, including route URL, function, and parameters.

- Views, including how views are called, location of view files, blade syntax, and template inheritance. What is the advantage of using template inheritance?
- How security measures to prevent cross-site scripting attacks and cross-site request forgery attack are implemented.

Task 5 – Implement PM example in Laravel

1. Make a copy of the partial implementation of assoc-*laravel* from *WebDev-Examples/week4* to your working (webAppDev) directory. Call this directory *week4/task5*
2. In console, go to *week4/task5* and execute the command: *composer update*

   (Composer update will fetch the library packages that were not pushed to GitHub due to .gitignore rules defined in Laravel).

3. Run the *task5* project to make sure it still works.
4. Complete the *task5* implementation by implementing:
   a. The two routes defined in routes/web.php
   b. The views for displaying search form and displaying the search results. These views should extend the same layout template. The layout template should be created under the views/layouts directory.
5. Implement the additional feature of displaying the user entered search string in the search results page. E.g. if user entered “John” then on the search results page it should display “Search result for John” on top of the search results.

Task 6 – Social Media in Laravel (Optional)

By using Laravel and template inheritance, Re-implement your Lab 1 exercise Social Media with Bootstrap solution.

Submitting your Laravel project

Before you submit a Laravel project, please remove the **vendor** directory before you zip your files. The vendor directory contains the library files used by the project, and it’s big. After you have submitted your work, you can restore the vendor directory by running the command: *composer update*