Web Programming 2022

Assignment #2: Basics of PHP Project

Deadline: 20-04-2022

Create a PHP project skeleton according to the following requirements:

Requirements:

1. The project should be structured as follows:

```
project/
   |--- asset/
   |      |--- css/ ← .css Bootstrap, Tailwind, Material or any other UI/CSS frameworks
   |      +--- js/  ← .js or UI/CSS/Theme related .js files
   |
   |--- layout/
   |      |--- header-css.php
   |      |--- footer-js.php
   |      |--- footer.php.                ← LAYOUT PARTIALS
   |      |--- navbar.php
   |      +--- sidebar.php(if exist)
   |
   |--- views/
   |      |--- contacts.php
   |      |--- array-doc.php              ← PAGES
   |      +--- string-doc.php
   |
   |--- config/
   |      |--- config.php(leave it empy)  ← CONFIG (DTABASE CONNECTION FOR LATER)
   |      |--- router.php
   |      +--- utility.php
   |
   |--- login.php
   |--- index.php ← Application Entry Point (composed from layout partials & view page)
   +--- signup.php
```

2. This project should be integrated with a one of CSS-Frameworks such as: Bootstrap, TailwindCSS, Material Design, or any other Themes based on above CSS frameworks (Please do not use plain HTML instead use themed components).
3. **login.php**: create a page for Login in the root folder of the project as shown above that contains a login form (Email & Password) with a link to Signup page.
4. **signup.php**: create a Signup page in the root folder of the project as shown above that contains a signing up form with three fields: Full name, Email and Password.
5. **index.php** page which is an entry point of the application is composed from layout partials in **layout/** directory and renders pages in **view/** based on user's requests. For example, the below URLs will render their corresponding views:

#	URL	Rendered view
1	http://localhost/project/index.php?p=contacts	contacts.php
2	http://localhost/project/index.php?p=array-doc	array-doc.php
3	http://localhost/project/index.php?p=string-doc	string-doc.php

Note: You should implement (routing technique) for mapping the **p=view-name** from URL to **view-name.php** file in **views/** directory for dynamic view loading (rendering) in **config/router.php** file.

View Pages: The view pages will contain the following contents:

Page 1 (contacts.php): This page will show the list of contacts as a table. To print an array of contacts as a table write a PHP function called **`render_table($a)`** in a separate file called **utility.php**, which takes a multi-dimension associative array as an argument and then renders the array as an HTML table.

Example1:

Input:
```
$input_array = [
   ["name"=>"Zara Azad", "phone"=>"+964000110011", "email"=>"zara@mail.com"],
   ["name"=>"Ahmed Rafiq","phone"=>"+964000223311","email"=>"arafiq@mail.com"],
   ["name"=>"Darbaz Khudhr","phone"=>"+964000440444","email"=>"darbaz@mail.com"],
];
```

Output:

#	Name	Phone	Email
1	Zara Azad	+964000110011	zara@mail.com
2	Ahmed Rafiq	+964000223311	arafiq@mail.com
3	Darbaz Khudhr	+964000440444	darbaz@mail.com

Page 2 (array-doc.php): This view page gives an appropriate example for each of the following array functions, for each function print the function name, the input value(s), and print the outputs.

1. is_array()
2. in_array()
3. array_merge()
4. array_keys()
5. array_key_exists()
6. array_shift()
7. array_push()
8. array_pop()
9. array_values()
10. array_map()
11. array_slice()
12. array_unshift()
13. array_rand()
14. count()
15. implode()

Page 3 (string-doc.php): Give an appropriate example for each of the following string functions, print the input value(s) and print the outputs in a file called **problem2.php**:

1. `strlen()`
2. `substr()`
3. `strpos()`
4. `str_split()`
5. `str_replace()`
6. `str_repeat()`
7. `strrev()`
8. `str_pad()`
9. `trim()`
10. `number_format()`
11. `chr()`
12. `ord()`
13. `addslashes()`
14. `strip_tags()`
15. `stripcslashes()`
16. `htmlentities()`
17. `html_entity_decode()`
18. `htmlspecialchars()`
19. `nl2br()`
20. `str_word_count()`
21. `explode()`
22. `md5()`
23. `sha1()`
24. `crypt()`

Note: the examples on page 2&3 should be in the following format. For printing arrays create a function called **dump()** in **utility.php** that uses **print_r()** internally wrapped with **"<pre>"** tag.

1. `strlen()`
Inputs: P1: `"Hello World";`
Output: `11`

Notes (Future Guidelines):

1. Please prepare this structure very carefully since you will be using this structure and theme for the upcoming assignment which is about **database connection**, **database CRUD operations**, and session-based login system.

2. If you understand the above structure, it will help you to understand the PHP framework structures in general and specifically Laravel, despite having many differences between this structure and PHP frameworks.

3. There may be a better way of organizing the PHP projects but this one is fine for beginners. When we deal with the database, we will create two other directories. The first one is **models/** directory that contains model classes for database operations. The second one is **controllers/** directory contains controller classes each class contains different action methods that represent user actions for (CRUD) operations.

4. **CRUD = Create Read Update Delete**