PhP unit

28/1/2020

Overview

In this project, we learn to work with phpunit. A php unit is deployed to debug a php program. It is a program oriented testing program. Testing is one of the most important concepts of extreme programming. It is important to carry out a professional project in which your code is covered by tests and that as far as possible you use the TDD methodology.

Objectives of project

- Understand what PHPUnit is and know its basics
- Improve your knowledge in PHP
- Improve your knowledge in testing
- Improve your programming knowledge

Table of Contents

Project requirements

- You must have a clear directory structure to be able to implement the pattern
- Create a clear and orderly directory structure
- The main page should show the URL's links with which you can navigate to the other controllers
- You must create a main controller that is responsible for receiving the request from the main page
- Both the code and the comments must be written in English
- Use the camelCase code style to define variables and functions
- In the case of using HTML, never use online styles
- Remember that it is important to divide the tasks into several sub-tasks so that in this way you can associate each particular step of the construction with a specific commit
- For the project documentation a PDF version is required within the repository
- You should try as much as possible, each commit is associated with a task
- Delete files that are not used or are not necessary to evaluate the project
- You must create a correctly documented README file in the root directory of the project (see guidelines in Resources)

Risk Management

Every project has risks. This risks must be taken into account to improve the workflow of the project. Managing these risks is important for due completion of project. Unchecked risks could not only lead to hamper of project deliverance but also a badly executed project. The risks hence associated with the project are documented as follows:

Risk	Risk level
Unfamiliarity with Phpunit	Medium to High
Designing the folder hierarchy	Medium to Low
Running the php tests	High
Delivering in time	Medium
Completing all the project requirements	Medium

Task Management

The following are the tasks which needed to be accomplished for the completion of the project. The tasks have been divided according to the project specifics. Each tasks have been clearly defined and their priority as well as their difficulty and time needed. Difficulty level is explained on a scale of 1 to 5 (5 being the most difficult). Priority is explained on the level of 1 to 5 (again 5 the highest parameter being the most prioritized work).

Task	Priority	Description	Difficulty	Time
Read the description of the project	5	This task involves complete understanding of requirements for the project	1	30 min
Create git repo	5	Creating of git repository for project execution and delivery	1	2 mins
Create files for the project	4	Creating the necessary files, folders and subfolders for this project. This is important especially for this project of PhP unit	1	5 mins
Download composer	3	Downloading the composer which is required for phpunit installation	2	10-20 mins
Install php unit	4	Installing the Php unit as stated in the website	3	30 mins
Create 2 functions as required	2	Create the 2 functions spaceConverter and lowerStrngConverter. These are stored in Util.php	4	1 hr
Run the tests	2	Run the tests from utiltests.php for test_slugs	5	2-3:30 hr
Documentation	1	Write the documentation and push it to github	2	2 hr

Defining this part is crucial to the development of the project. It is important to make a good analysis of the situation to organize the project in a good way.

Chronogram

This shows the time-line it took for the project to finish and also the tasks that were achieved during those timeline.

The tasks

Tasks	Mon 1000 hrs	Mon 1100 hrs	Mon 1200 hrs	Mon 1300 hrs	Mon 1400 hrs	Mon 1500 hrs	Tue 1000 hrs
Read project descriptio-n	**X**						
Git Repo	**X**						
Create files	**X**						
Download composer		**X**					
Install PhP unit		**X**					
Create 2 functions			**X**				
Run the tests				**X**	**X**		

Documentation						X	X

Git Workflow

For this project, all commits we'll be pushed directly to the Master branch. All commits will use a descriptive message, so that the admin and other internet user can follow through the processes.
For more information go to :
https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

Incidents

- None

Technologies used

For this project, we will use the following technologies:

- HTML
- CSS & Bootstrap
- Php
- Php unit