Composer

Overview

In this project, we learn the usage of composer. Composer is a tool that manages the dependencies of our PHP project. Thanks to this tool, we can work with the installation of those libraries that our project needs in a comfortable way, avoiding all the problems involved in installing the dependencies manually and without any control or supervision. The dependencies are installed in a directory (default called / vendor). We have to explore the dependencies of project using the composer dependency manager

Objectives of project

- Understand that it is a dependency manager.
- Understand what Composer is and what its fundamentals are.
- Improve your knowledge in PHP in professional work environments

Table of Contents

1. Overview
2. Objectives
3. Project requirements
4. Risk management plan
5. Tasks for the project
6. Chronogram
7. Calendar
8. Git Workflow

9. Technologies used
10. Incidents

Project requirements

- Install composer to start importing dependencies in your project. You can install composer locally or globally. Choose the option that you consider most appropriate.
- Execute the necessary operations to install the dependencies on which your project depends. It is important that you check if your dependence is for the development environment or for the production environment, since depending on this point the installation method will require different parameters to be executed.
- Finally, verify that your dependencies have been installed and can be used from your project without any problem.
- Find out how you can update a dependency on your project
- Research how you can eliminate a dependency from your project
- Investigate what the composer.json file is and what it is for
- The first source of this project is to prepare a list of what units your project needs. To use as a dependency a library called Guzzle. This library is not the core of the project, it is simply used as a reference to put the potential of Composer into practice and how to use it correctly in your projects. You can use other libraries if you consider it necessary, in fact, we recommend it.

Risk Management

Every project has risks. These risks must be taken into account to improve the workflow of the project. Managing these risks is important for due completion of the 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 Composer	High
Unfamiliarity with Guzzle	Medium
Health issues	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 task has 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
Install Composer	4	Write the XML file as required	2	10-20 mins
Install Guzzle	3	Answer all the questions asked in the project requirements	1	10-20mins
Check dependencies by writing a script	2	Write the guzzle script in the index.php	3	20 mins
Documentation	1	Write the documentation and push it to github	2	30 mins

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	Wed 0900 hrs	wed 1000 hrs
Read project descriptio-n	X	
Git Repo	X	
Create files	X	
Download Composer	X	
Download Guzzle		X
Write the script		X
Documentation		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 users 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:

- Composer
- Php
- Guzzle
- GIT