ASSESSMENT DETAILS			
Number & Title:	**AS3 Lab test: Practical (Group)**		
Weightage:	20%	Release Date:	26 February 2025
Assessment Type:	Assignment	Due Date:	9 April 2025
Module Lecturer Email:	bahit.hamid@pb.edu.bn		
Minimum Required Hours:	20 hours		

ASSESSMENT OBJECTIVES

The objective for this assessment: setup, configures and deploy a Virtual Private Server (VPS) that will run a PHP based framework website on a cloud service provider through a container system such as Docker. Setup the VPS according to the specification that is required and have an Internet reachable web server.

TASK SUMMARY

1. This is **group-based** assessment where students will work together using their own laptop/machine to solve a deployment of a PHP web application through the use of containerization.
2. The summary of add-ons to this assignment:
    a. Deploy a Laravel Web Application with supporting servers (containers) using Docker or equivalent.
    b. Configure and setup necessary files and configuration on both Laravel Web Application to work with Docker.
    c. Produce a Docker-compose file that contains the necessary settings for Laravel Web Application to run in a Docker environment.
3. To complete the tasks below:
    a. **Task 1: Submission of Docker files and Docker Compose File – 5% marks**
    b. **Task 2: Demonstration of running Docker on VPS (practical) – 15% marks**
4. To refer to **Academic Policy Assignment Rules** (posted on LMS).

INSTRUCTIONS

Read the following description:

Task 1: Submission of Docker files and Docker Compose File – 5% marks

Assuming you have installed a Docker instance in the machine you are working (both **`docker`** and **`docker compose`** are available in the command line), you are expected to do the following:

Have a running Laravel Application (default, no changes to source code) with a **customized .env** file running in a Docker container.

MySQL server running on Docker container that is used by Laravel web application above.

Redis server running on Docker container that is used by Laravel web application above.

Task 2: Demonstration – 25% mark

Demonstration session for finished Laravel web application on Docker [Group]

You are expected to show your Docker based Laravel web application server in a 10-minute demonstration, where you are to show evidence that you have a live running Laravel web application that was deployed using docker by using **`docker compose up`** and **`docker compose down`** and had implemented the configuration requirements set in this assignment.

The invigilator will follow a predefined list of groups in each timetable and will follow through a checklist of evidence that you will need to demonstrate. This means, you will need to know all the commands needed to show the configuration you have used.

The invigilator will sit next to you with your group's appointed laptop on showing mainly two windows, a browser for showing the web server is running with the predefined website and a console / terminal window.

Students will only be awarded points for each requested evidence. Failure to show evidence for a given requirement will be taken note of.

Group customization:

To allow unique implementation of each group for this assignment, each group will be given a group name. E.g., from web module students; dwtygroup1 and for application development students; dadtgroup1.

The grouping will be given on LMS. Members of each group must use their group name in customizing their Docker implementation. In the grading criteria, whenever the syntax <group> is stated, you must use your group name.

The detail tasks are as follow:

1. The customization needed in the on Laravel Docker Container:
    a. Using **Ubuntu 24.04** image
    b. Docker maintainer: "**<group>**"
    c. Time zone **"Asia/Brunei"**
    d. PHP **8.3**
    e. Nginx
    f. Composer
    g. Use **port 8080** on localhost side.
    h. Laravel installed in **/var/www/html** directory inside the docker container.
2. Use Docker to network all components required by Laravel: **MySQL** and **Redis**.
3. Have a **local file structure** which has Laravel source code to be used as a shared volume with Laravel Docker container.

On the day of submission, each student must submit a zip file containing **`docker-compose.yml`** as well as other configuration settings (that is used to customize the servers in the containers) in creating the Laravel web application. Additional instruction may be given at the LMS.

If you have any question regarding this assignment, you are free to submit a forum post on LMS. Students however are expected to do their own research to implement a VPS to the specification required by this assignment.

Mid Assessment Compulsory Check – 3% Weightage

You will need to submit the required details by **11:59PM, Sunday, 16 March 2025** on WY4205 Development Operations LMS page under Assessment 3 section with an activity named "**Mid Assessment Check**" under AS3 section.

This submission of details will have a weighted mark of 3% for this assessment.

In part with student progress check-up with the assignment, students need to upload there:

- IP Address of their VPS
- The username of the non-root user account
- Upon ssh into the VPS, Docker is installed: docker and docker-compose are available inside the machine to be used and run.

Upon opening student VPS's IP address in a web browser, it should show a website running served by your web server. Students also need to put the following public key of my own SSH key into the non-root user account authorize_keys so I will be able to verify your VPS by logging in manually:

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAII10iQdtZ+BwsKj7GXcAJ184/bV3NhIWtqdSR4X5CmRr bahit.hamid@gmail.com

Demonstration Timetable

The timetable of the demonstration day, which will on the **week 14**. We will use our DevOps module time slots of the week to conduct the demonstration. This will be first come first serve basis. Invigilator will use the student's list for each group for reference and queue.

Tutorial in Deploying Docker for Laravel Web Application

The tutorial your group should be following for this assignment can be found on Servers for Hackers website:

- https://serversforhackers.com/s/docker-in-dev-v2-i
- https://serversforhackers.com/s/docker-in-dev-v2-ii

The above video tutorials contain all the requirements you'll need for your AS3. However, the video series will not cover installation of Docker into the machine or VM you'll be working on. For instruction on how to install Docker can be found on LMS.

Although the video series contains step-by-step instruction of deploying a Laravel web application on Docker (with other supporting servers in different containers networked together in Docker), you'll need to customise part of the commands and instruction to fit this assignment's requirement (see first section of this assignment).

DELIVERABLES

Upon submission, students must submit the task 1 in **softcopy** via LMS.

- 2 files:
  - docker file.
  - docker-compose.yml file
- It is the student's responsibility to ensure no aspect of their work is plagiarised or the result of other unfair means. **Any plagiarised work by default will result in failure to comply with '0' or 'Fail'**.

Submission link will be share in due time in LMS.

IMPORTANT: Due to the nature of this assignment where the solution for the practical can be practiced among other peers, your logbook must be uniquely yours – any attempt to plagiarise work will be dealt according to school policy on the matter.

Students who are found having **similar design and structure of logbook will be penalised as plagiarism**.

GRADING CRITERIA

Below is the guideline on how this assessment is being evaluated:

	Criteria		Percentage
Task 1: Docker configuration files and Docker Compose File [Total 5% marks]			
1.	Docker Compose File	Contains all the necessary settings for: • Spinning up the necessary **services**, **networks** and **volumes.** • **Have a presistent volume** to be used by Laravel and MySQL. • **Network that connects all the containers.**	2.5
2.	Docker Compose File: database	Custom: • environment settings for **root password**, • **DB name** • **DB user** • **DB user's password**	0.5
3.	Docker Compose File: Application Port and Working Directory	• Use port **80** on Docker side and use port **8080** localhost side. • Use a working directory in local machine called '**laravel'**	0.5
4.	Docker Compose File: Network	• Use a network name shared by all containers called **sysops**.	0.5
5.	Docker Compose File: Services	• Laravel container should be called '**<group>app'** • Redis container should be called '**rediscontainer'** • Database container should be called '**mysqldbcontainer'**	0.5
6.	Docker Compose File: Volumes	• Database volume used by MySQL should be called '**mysqldbdatavolume'** • Caching volume used by Redis should be called '**redisdatavolume'**	0.5
Mid Assessment Compulsory Check [Total 3% marks]			
1.	• IP Address of VPS given and reachable via ping command		1
2.	• Username of non-root user given and able to login via ssh via provided SSH key		1
3.	• Command: **docker** and **docker compose** are available in terminal		1

Task 2 Demonstration [Total 12% marks]			
1.	Docker available on VPS / local machine	• docker is available in command line • docker service is running	1
2.	Docker compose available on VPS / local machine	• docker compose is available in command line	1
3.	Docker compose / Docker spins up Laravel web App	• docker compose/docker can run script • docker compose/docker runs script with no error	1
4.	Laravel web application accessible through browser via port 8080	• Website is running and shows on browser • Website is accessible on port 8080	1
5.	Dockerfile shows maintainer to be modified using the correct respective <studentid>	• In dockerfile shows the maintainer is indeed in fact the individual name	1
6.	Have an existing .env file that contains the working settings to be used in container	• Laravel was setup correctly and .env shows the required settings	1
7.	Application folder (Laravel) is found on local machine / VPS and used as working directory	• A folder named "Laravel" is used as source code and used by container • This folder owner is not 'root' and owned by a non-root user	1
8.	Container's time zone set to Brunei's time zone.	• When inside the container's environment, indeed it shows +8 time zone. • The container was named using individual naming scheme	1
9.	Container's using PHP 8.3	• Evidence shows that the container is showing PHP is running • Evidence shows that the container is showing PHP is version 8.3.	1
10.	Laravel installed inside /var/www/html	• The content of html folder is the Laravel source code	1
11.	Existing .env file is setup to use Redis with correct hostname found in docker-compose.yml	• Inside the .env, Redis is correctly setup with docker's redis server • Redis service should be called 'rediscontainer' • Redis credential matches container's credential	1
12.	Existing .env file is setup to use MySQL with correct hostname found in docker-compose.yml	• Inside the .env, MySQL is correctly setup with docker's MySQL server • Database service should be called 'mysqldbcontainer' • Database credential matches container's credential	1

Total %:	20

- END OF ASSIGNMENT BRIEF -