

5. Контейнеризація.

Контрольні запитання

- 04.01. Що таке контейнеризація?
- 04.02. Що таке контейнер?
- 04.03. Що таке Docker?
- 04.04. Що таке образ (image) в контексті Docker?
- 04.05. Що таке сховище (volume) в контексті Docker?
- 04.06. Для чого потрібен Dockerfile?
- 04.07. Що таке шар (layer) в Dockerfile? Які команди створюють новий шар?
- 04.08. Для чого використовується інструмент Docker Compose? Яким файлом описується побудова конфігурації для Docker Compose?

Завдання для аудиторної роботи

[A]04.01. Встановити Docker Desktop.

- Запустити найпростіший контейнер

```
docker run hello-world
```

- Перевірити усі наявні контейнери

```
docker ps -a
```

- Зупинити контейнер

```
docker stop <container_id>
```

- Видалити контейнер

```
docker rm <container_id>
```

[A]04.02. Описати і запустити файл **docker-compose.yml** для створення контейнерів з наступною конфігурацією:

1. База даних PostgreSQL із зовнішнім сховищем (volume);
2. Веб-додаток pgAdmin під'єднаний до PostgreSQL;
3. Простий застосунок на Python, який при запуску створює в базі даних таблицю (якщо її ще не існувало) та додає туди пару записів. Також показує повний вміст цієї таблиці.

Змінні оточення (користувачі бази даних, паролі тощо) повинні зберігатися в файлі **.env**.

Завдання для самостійної роботи

[B]04.01. Описати і запустити файл **docker-compose.yml** для створення контейнерів з наступною конфігурацією:

1. База даних MySQL із зовнішнім сховищем (volume);
2. Веб-додаток phpMyAdmin під'єднаний до MySQL;
3. Простий застосунок на Python, який при запуску створює в базі даних таблицю (якщо її ще не існувало) та додає туди пару записів. Також показує повний вміст цієї таблиці.

Змінні оточення (користувачі бази даних, паролі тощо) повинні зберігатися в файлі **.env**.