



DEV CHALLENGE 1101

Backend Task | Online Round

Level: **Standard / Pro**

Duration = **4 hours**

Deadline: **14.10.2018**

1. Завдання:

Написати систему, яка гарантуватиме безпеку передачі паролів від однієї людини іншій. Вибір технологій для виконання завдання не обмежується і лягає на плечі учасника.

Вхідні дані: створити сервіс для шарінгу (обміну) паролів, який буде приймати на вхід пароль в чистому вигляді і повертати посилання.

Вимоги:

- посилання має завжди бути унікальним, навіть для однакових паролів
- посилання має обмежений час дії
- посилання може бути використане лише один раз
- в разі доступу злоумисника до бази даних це не має викрити збережених паролів

Очікуваний результат:

Серверний API з базовою функціональністю:

- зашифрувати пароль, відповідає посиланням
- перехід за посиланням повертає розшифрований пароль

2. Формат представлення результатів

2.1. Рішення завантажувати в особистому кабінеті на сайті в **ОДНОМУ** файлі-архіві з назвою у форматі **Ім'я Прізвище.zip**.

2.2. README файл, в якому обов'язково вказувати:

- 2.2.1.** Кроки для старту сервісу
- 2.2.2.** Кроки для запуску тестів
- 2.2.3.** Опис API
- 2.2.4.** Методологію, яку ви обрали, та пояснення, чому, на ваш погляд, вона найкраще підходить.
- 2.2.5.** Речі, на які ви б хотіли звернути увагу або наступні кроки для вдосконалення вашого сервісу.

3. SCALEME файл, в якому описано, як ваша система масштабується, які для цього необхідні конфігурації. Зверніть увагу, що назва архіву - єдине місце, де ви вказуєте свої персональні дані. Назви файлів всередині архіву не мають містити вашого імені, прізвища чи нікнейма. Розмір архіву з рішенням не має перевищувати 10 MB.

УВАГА! Передбачається використання останніх версій програмного забезпечення на момент здачі роботи. Після запуску проект повинен бути доступний за адресою - <http://localhost:8080/>. Обов'язково перевіряйте запуск вашого проекту на іншому комп'ютері перед відправкою. Проект повинен запускатися однією командою (на вибір учасника):

- «vagrant up» (Vagrant & Virtual Box)
- «docker-compose up» (Docker & Docker Compose).

4. Організатори та судді залишають за собою право дискваліфікувати роботу учасника, якщо робота:

- 4.1.** Містить будь-яку вказівку на ім'я, прізвище, електронну пошту, компанію, адресу чи інші персональні дані учасника
- 4.2.** Виконана у іншому форматі, ніж вказано у завданні
- 4.3.** Виконана за допомогою сторонніх осіб, а не учасником особисто



3. Обмеження та критерії оцінювання

Категорія	Критерії	Бали
STANDARD	Працездатність додатку	100
STANDARD	Дотримання кодстайлу, структурованність та простота коду	30
STANDARD	Вибір та архітектура бази даних	50
BONUS	Наявність юніт тестів	24
BONUS	Наявність інтеграційних тестів	24
BONUS	Наявність документації, на кшталт API Doc	16
PRO	Можливість одночасно обробляти декілька паролей	19
PRO	Можливість вказати час дії посилання	19
PRO	В базі даних наявна історія створення і запиту паролей	19
PRO	Окремо зберігаються переходи за посиланням і можна вилучити по запиту підозрілі переходи	19

4. Контакти

4.1. Рішення необхідно завантажити у особистому кабінеті на сайті devchallenge.it за обраною номінацією до 14 жовтня, 23:59 (EEST). Після вичерпання часу можливість завантажити роботи на сайт буде автоматично заблокована.

4.2. Учасник має право повторно відправити своє рішення завдання до кінця дедлайну, але оцінюватися буде останній надісланий варіант.

4.3. Питання та уточнення щодо змісту завдання ви можете задати, заповнивши форму зворотнього зв'язку на сайті, вказавши назву номінації.

4.4. Організаційні запитання надсилайте на пошту team@devchallenge.it

4.5. Оголошення фіналістів відбудеться 30 жовтня.