

Zadanie Python (dowolny framework i narzędzia):

Projekt i implementacja samodzielnego serwisu eMenu, służącego jako restauracyjna karta menu online.

Wymagania stawiane aplikacji:

API niepublicznie:

1. REST API do zarządzania menu
2. Możliwość tworzenia wielu wersji kart (menu) o unikalnej nazwie.
3. Każda karta może zawierać dowolną liczbę dań.
4. Każde danie powinno charakteryzować się: nazwą, opisem, ceną, czasem przygotowania, datą dodania, datą aktualizacji, informacją czy danie jest wegetariańskie
5. Każda karta charakteryzuje się: nazwą (unikalna), opisem, datą dodania, datą aktualizacji
6. API musi być zabezpieczone przed nieautoryzowanym dostępem (po autoryzacji użytkownika)

API publicznie:

1. Rest API do przeglądania niepustych karta menu.
2. Możliwość sortowanie listy po nazwie oraz liczbie dań, za pomocą parametrów GET
3. Filtrowanie listy po nazwie oraz okresie dodanie i ostatnie aktualizacji
4. Detal karty prezentujący wszystkie dane dotyczące karty oraz dań w karcie.

Raportowanie:

1. Przygotować mechanizm, który raz dziennie o 10:00 wyśle e-mail do wszystkich użytkowników aplikacji
2. E-mail musi zawierać informację o nowa dodanych przepisach oraz ostatnio zmodyfikowanych przepisach
3. Wysyłamy informację tylko o tych, który zostały zmodyfikowane poprzedniego dnia.

Dodatkowo:

1. Konieczne jest załączenie instrukcji instalacji oraz uruchomienia projektu
2. Mile widziane jest przygotowanie aplikacji po uruchomienie w Docker (Dockerfile oraz docker-compose.yml do uruchomienia aplikacji)
3. Dopuszczalne jest korzystanie z ogólnodostępnych rozwiązań.
4. Dane inicjalizacyjne do projektu są mile widziane.
5. Konieczne jest udokumentowanie API za pomocą Swagger lub innego narzędzia (dokumentacja powinna być generowana automatycznie)
6. Możliwość dodania zdjęcia dania nie jest wymagana, lecz jej obecność zostanie pozytywnie odebrana.
7. Sposób dostarczenia aplikacji jest dowolny, jednak w miarę możliwości zachęcamy do skorzystania z GitHub-a.
8. Dostarczony kod powinien posiadać pokrycie testami na poziomie min. 70% (coverage), dotyczy wyłącznie kodu napisanego przez kandydata (bez uwzględniania testów zewnętrznych bibliotek).

9. Należy pamiętać o odpowiednich ustawieniach lokalizacyjnych oraz problemach związanych z optymalizacją liczby zapytań do bazy danych.
10. Koniecznym jest wykorzystanie relacyjnego silnika bazodanowego (możliwe do uruchomienia na PostgreSQL bez ingerencji w kod, prócz konfiguracji)

Oceniane będą:

1. wykorzystanie i znajomość języka Python
2. wykorzystanie i znajomość wybranego Frameworka,
3. wykorzystanie i znajomość dobrych praktyk pisania kodu
4. wykorzystanie i znajomość wzorców projektowych
5. wykorzystanie i znajomość standardów REST API
6. wykorzystanie i znajomość systemu kontroli wersji GIT